

Troubleshooting Internetworking Systems



Troubleshooting Internetworking Systems

Software Release 9.1

Corporate Headquarters:
1525 O'Brien Drive
Menlo Park, California 94025 USA
1-415-326-1941
1-800-553-NETS

Customer Order Number: DOC-TIS9.1
Text Part Number: 78-1077-01



The products and specifications, configurations, and other technical information regarding the products contained in this manual are subject to change without notice. All statements, technical information, and recommendations contained in this manual are believed to be accurate and reliable but are presented without warranty of any kind, express or implied, and users must take full responsibility for their application of any products specified in this manual. THIS MANUAL IS PROVIDED "AS IS" WITH ALL FAULTS. CISCO DISCLAIMS ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING THOSE OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, OR ARISING FROM A COURSE OF DEALING, USAGE OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL OR INCIDENTAL DAMAGES, INCLUDING WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Some states do not allow limitation or exclusion of liability for consequential or incidental damages or limitation on how long implied warranties last, so the above limitations or exclusions may not apply to you. This warranty gives Customers specific legal rights, and you may also have other rights that vary from state to state.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright (c) 1981 Regents of the University of California.

Redistribution and use in source and binary forms are permitted provided that the above copyright notice and this paragraph are duplicated in all such forms and that any documentation, advertising materials, and other materials related to such distribution and use acknowledge that the software was developed by the University of California, Berkeley. The name of the University may not be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Point-to-Point Protocol. Copyright (c) 1989 Carnegie Mellon University. All rights reserved.

Redistribution and use in source and binary forms are permitted provided that the above copyright notice and this paragraph are duplicated in all such forms and that any documentation, advertising materials, and other materials related to such distribution and use acknowledge that the software was developed by Carnegie Mellon University. The name of the University may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Notice of Restricted Rights:

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c) of the Commercial Computer Software - Restricted Rights clause at FAR § 52.227-19 and subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS § 252.227-7013.

The information in this manual is subject to change without notice.

APPI, ciscoBus, Cisco Systems, CiscoWorks, CxBus, Netscape, The Packet, and SMARTnet are trademarks, and the Cisco logo is a registered trademark of Cisco Systems, Inc.

All other products or services mentioned in this document are the trademarks, service marks, registered trademarks, or registered service marks of their respective owners.

Troubleshooting Internetworking Systems

Copyright © 1993, Cisco Systems, Inc.

All rights reserved. Printed in USA.

Table of Contents

About This Manual xxvii

- Audience and Scope xxvii
- Document Organization and Use xxvii
- Document Conventions xxix
- Related Documentation xxix

Service and Support xxxi

- Warranty Information xxxi
- Maintenance Agreements xxxi
- Customer Support xxxii

Obtaining Additional Information xxxiii

- Ordering Additional Cisco Publications xxxiii
- Obtaining Cisco Technical Information Electronically xxxiii
- Obtaining Information from Other Sources xxxiv
 - Obtaining RFCs xxxiv
 - Obtaining Technical Standards xxxvi

Chapter 1 Troubleshooting Overview 1-1

- Focus on Symptoms, Causes, and Actions 1-1
 - What This Guide Is Not 1-2
- Using This Publication 1-3
 - General Problem-Solving Model 1-3
 - Symptom Modules 1-5
 - Troubleshooting Scenarios 1-6
 - Using This Publication to Troubleshoot Specific Symptoms 1-6
 - Using This Publication as a Tutorial 1-6
 - Using This Publication with Other Cisco Publications 1-7
- Using Cisco Diagnostic Tools 1-8
 - Using Show Commands 1-8
 - Using Debug Commands 1-9

Using Ping and Trace Commands	1-9
Using Core Dumps	1-9
Diagnosing Cisco Hardware	1-10
Physically Inspecting Your System	1-10
Applying Power and Evaluating the System	1-11
Testing and Verifying Operation	1-13
Using CiscoWorks to Troubleshoot Your Internet	1-17
Using CiscoWorks to Troubleshoot Connectivity Problems	1-17
Using CiscoWorks to Troubleshoot Performance Problems	1-18
Using Third-Party Troubleshooting Tools	1-19
Troubleshooting Media Problems	1-20

Part One ***Troubleshooting Connectivity***

Chapter 2 ***Connectivity Problem Scenarios 2-1***

Connectivity Scenario Overview	2-2
Apple Service Availability Scenario	2-3
Symptoms	2-3
Environment Description	2-4
Diagnosing and Isolating Problem Causes	2-5
Problem Resolution Process	2-6
Problem Solution Summary	2-12
Concurrent Routing and Source-Route Bridging Connectivity Problems	2-15
Symptoms	2-15
Environment Description	2-15
Diagnosing and Isolating Problem Causes	2-16
Problem Solution Summary	2-19
Translational Bridging, SRT, STUN, and SDLLC Connectivity Problems	2-21
Symptoms	2-21
Environment Description	2-21
Diagnosing and Isolating Problem Causes	2-23
Problem Solution Summary	2-27

Novell Network Server Connectivity Scenario	2-33
Symptoms	2-33
Environment Description	2-34
Diagnosing and Isolating Problem Causes	2-34
Problem Solution Summary	2-42
TCP/IP Route Redistribution and Access Control Scenario	2-45
Symptoms	2-45
Environment Description	2-46
Diagnosing and Isolating Problem Causes	2-46
Problem Solution Summary	2-49
X.25 WAN Router Initial Installation Problems	2-51
Symptoms	2-51
Environment Description	2-51
Diagnosing and Isolating Problem Causes	2-52
Problem Solution Summary	2-61

Chapter 3

Troubleshooting Apple Connectivity 3-1

AppleTalk Internetworking Terminology	3-1
Networks and Internets	3-1
Phase 1 and Phase 2 Routers	3-2
Nonextended and Extended Networks	3-2
AppleTalk Internetworking Diagnostic Tips	3-3
Common AppleTalk Internetworking Problems	3-3
Preventing AppleTalk Configuration Problems	3-7
Common AppleTalk Problem Diagnostics	3-10
AppleTalk Connectivity Symptoms	3-12
Symptom Summary	3-12
Users Cannot See Zones or Services on Remote Networks	3-13
Possible Causes and Suggested Actions	3-13
Services on a Network Not Visible to Other Networks	3-14
Possible Causes and Suggested Actions	3-14
Users Cannot Access Services on Remote Networks	3-16
Possible Causes and Suggested Actions	3-16
Some Zones Missing from Macintosh Chooser	3-18
Possible Causes and Suggested Actions	3-18

Services Not Always Available; Fade In and Out	3-20
Possible Causes and Suggested Actions	3-20
Services Visible, but Users Cannot Connect	3-22
Possible Causes and Suggested Actions	3-22
Zone List Changes Each Time Chooser Is Opened	3-23
Possible Causes and Suggested Actions	3-23
Connections to Services Drop	3-24
Possible Causes and Suggested Actions	3-24
Port Seems Stuck in Restarting or Acquiring Mode	3-25
Possible Causes and Suggested Actions	3-25
Old Zone Names Still Appear in the Chooser	3-26
Possible Causes and Suggested Actions	3-26

Chapter 4

Troubleshooting IBM Connectivity 4-1

Diagnosing IBM Network and Token Ring Problems	4-2
IBM Network and Token Ring Connectivity Symptoms	4-3
Routing Does Not Function in SRB Environment	4-4
Possible Causes and Suggested Actions	4-4
Routing in SRB Network Fails Unexpectedly	4-5
Possible Causes and Suggested Actions	4-5
No Communication over SRB	4-6
Possible Causes and Suggested Actions	4-6
Blocked Communication over Remote SRB	4-8
Possible Causes and Suggested Actions	4-8
Intermittent Communication Failures over Remote SRB	4-9
Possible Causes and Suggested Actions	4-9
Users Cannot Communicate over Cisco Translational Bridge	4-10
Possible Causes and Suggested Actions	4-10
Traffic Cannot Get Through Router Implementing SRT	4-12
Possible Causes and Suggested Actions	4-12
Users Cannot Make Connections over Router Configured for SDLLC	4-13
Possible Causes and Suggested Actions	4-13
IBM RS-232 Signaling Requirements Summary	4-14
Preventive Actions in SDLLC Environments	4-15
Intermittent Connectivity over Router Configured for SDLC	4-16
Possible Causes and Suggested Actions	4-16

Router Is Unable to Connect to Token Ring	4-17
Possible Causes and Suggested Actions	4-17
Router Is Not Communicating with IBM SDLC Devices over RS-232	4-19
Possible Causes and Suggested Actions	4-19
SDLC Sessions Fail over Router Running STUN	4-20
Possible Causes and Suggested Actions	4-20
NetBIOS Devices Cannot Communicate over Remote SRB	4-22
Possible Causes and Suggested Actions	4-22
Router Cannot Be Linked from LAN Network Manager	4-23
Possible Causes and Suggested Actions	4-23

Chapter 5

Troubleshooting Novell Connectivity 5-1

Novell IPX Internet Diagnostic Overview	5-1
Problem Isolation in Novell IPX Networks	5-2
Novell Internetworking Connectivity Symptoms	5-3
Symptom Summary	5-3
Clients Cannot Communicate with NetWare Servers over Router	5-4
Possible Causes and Suggested Actions	5-4
SAP Updates Not Getting Through Router	5-9
Possible Causes and Suggested Actions	5-9
Novell NetBIOS Packets Cannot Get Through Router	5-11
Possible Causes and Suggested Actions	5-11
Helper Address Specification Hints	5-13
Clients Cannot Connect to Server over Packet-Switched Network	5-21
Possible Causes and Suggested Actions	5-21
Notes About PSN Address Map Specifications	5-22

Chapter 6

Troubleshooting TCP/IP Connectivity 6-1

TCP/IP Internet Diagnostic Overview	6-1
Problem Isolation in TCP/IP Networks	6-2
TCP/IP Connectivity Symptoms	6-3
Symptom Summary	6-3
Host Cannot Access Offnet Host(s)	6-4
Possible Causes and Suggested Actions	6-4
Host Cannot Access Certain Networks	6-6
Possible Causes and Suggested Actions	6-6

Connectivity Available to Some Hosts, but Not Others	6-7
Possible Causes and Suggested Actions	6-7
Some Services Are Available, Others Are Not	6-8
Possible Causes and Suggested Actions	6-8
Users Cannot Make Connections When One Path is Down	6-9
Possible Causes and Suggested Actions	6-10
Router Sees Duplicate Routing Updates and Packets	6-11
Possible Causes and Suggested Actions	6-11
Routing Works for Some Protocols, Not for Others	6-12
Possible Causes and Suggested Actions	6-12
Router/Host Cannot Reach Certain Parts of Its Own Network	6-13
Possible Causes and Suggested Actions	6-13
Traffic Is Not Getting Through Router Using Redistribution	6-15
Possible Causes and Suggested Actions	6-15

Chapter 7 ***Troubleshooting WAN Connectivity 7-1***

Diagnosing WAN and Serial Line Problems	7-1
Using the Show Interfaces Command to Troubleshoot Serial Lines	7-2
Deciphering Serial Line Status Diagnostics	7-3
Basic Serial Diagnostic Fields	7-6
WAN-Specific Diagnostic Fields	7-11
Using the Show Controllers Command to Troubleshoot Serial Lines	7-13
Using Debug Commands to Troubleshoot Serial Lines	7-13
Special Serial Line Tests	7-14
CSU/DSU Local and Remote Loopback Tests	7-15
Extended Ping Tests	7-16
Troubleshooting Clocking Problems	7-18
Adjusting Buffers to Ease Overutilized Serial Links	7-20
WAN and Serial Line Connectivity Symptoms	7-25
Intermittent Connectivity	7-26
Possible Causes and Suggested Actions	7-26
Connections Die as Load Increases	7-27
Possible Causes and Suggested Actions	7-27
Connections Die at a Particular Time of Day	7-28
Possible Causes and Suggested Actions	7-28

Connections Die After Some Period of Normal Operation	7-29
Possible Causes and Suggested Actions	7-29
Users Cannot Connect to Resources over New HDLC Link	7-30
Possible Causes and Suggested Actions	7-30
Users Cannot Connect to Resources over New X.25 WAN Link	7-31
Possible Causes and Suggested Actions	7-31
Users Cannot Connect to Resources over New Frame Relay Link	7-33
Possible Causes and Suggested Actions	7-33
Users Cannot Connect to Resources over New SMDS Link	7-35
Possible Causes and Suggested Actions	7-35
Some Users Cannot Connect to Resources over WAN	7-37
Possible Causes and Suggested Actions	7-37

Part Two
Chapter 8

Troubleshooting Performance

Performance Problem Scenarios 8-1

Performance Scenarios: Overview and List 8-2

Performance Problems in Novell IPX Internet After Bandwidth Upgrade 8-3

 Symptoms 8-3

 Environment Description 8-3

 Diagnosing and Isolating Problem Causes 8-4

 Problem Solution Summary 8-4

Performance Problems in Novell IPX Internet After Switch to Routing 8-5

 Symptoms 8-5

 Environment Description 8-5

 Diagnosing and Isolating Problem Causes 8-6

 Problem Solution Summary 8-6

Slow Novell IPX Performance over Router Connecting 16-Mbps Rings 8-7

 Symptoms 8-7

 Environment Description 8-7

 Diagnosing and Isolating Problem Causes 8-8

 Problem Solution Summary 8-8

Slow Novell Performance over Ethernet Backbone 8-9

 Symptoms 8-9

 Environment Description 8-9

Diagnosing and Isolating Problem Causes	8-10
Problem Solution Summary	8-10
Slow Novell Performance over Matching Parallel Links	8-13
Symptoms	8-13
Environment Description	8-13
Diagnosing and Isolating Problem Causes	8-14
Problem Solution Summary	8-14
Slow Novell Performance over Unequal Parallel Links	8-15
Symptoms	8-15
Environment Description	8-15
Diagnosing and Isolating Problem Causes	8-16
Problem Solution Summary	8-16
Poor Performance over TCP/IP Serial Network	8-17
Symptoms	8-17
Environment Description	8-17
Diagnosing and Isolating Problem Causes	8-18
Problem Resolution Process	8-18
Problem Solution Summary	8-20
Slow Host Response over 56-Kbps HDLC Link	8-21
Symptoms	8-21
Environment Description	8-21
Diagnosing and Isolating Problem Causes	8-22
Problem Resolution Process	8-22
Problem Solution Summary	8-27

Chapter 9	<i>Troubleshooting Internet Performance 9-1</i>
	Poor Internetwork Performance Symptoms 9-2
	Sporadic Service Availability and Poor AppleTalk Internet Performance 9-3
	Possible Causes and Suggested Actions 9-3
	Slow Performance and Intermittent Loss of Connections over RSRB 9-5
	Possible Causes and Suggested Actions 9-5
	Poor Novell Server Performance over Router in LAN Internet 9-6
	Possible Causes and Suggested Actions 9-6
	Poor Novell Server Performance over Router in WAN 9-7
	Possible Causes and Suggested Actions 9-7

Generally Slow Performance in TCP/IP Internetworks	9-8
Possible Causes and Suggested Actions	9-8
Slow TCP/IP Performance Despite Multiple Paths	9-9
Possible Causes and Suggested Actions	9-9
Slow Host or Network Response over WAN or Serial Link	9-11
General Diagnostic Information	9-11
Possible Causes and Suggested Actions	9-11
Loss of Connections over WAN or Serial Link	9-13
Possible Causes and Suggested Actions	9-13

Part Three Debug Command Reference

Chapter 10 Debug Command Reference 10-1

General Debugging Information	10-1
Using Debug Commands	10-1
Using the Debug ? Command	10-2
Using the Debug All Command	10-2
Generating Debugging Command Output	10-2
Redirecting Debugging and Error Message Output	10-3
Debug Command Listing	10-8
Debug Apple-ARP	10-9
Debug Apple-Errors	10-10
Debug Apple-Events	10-12
Debug Apple-NBP	10-16
Debug Apple-Packet	10-19
Debug Apple-Routing	10-21
Debug Apple-ZIP	10-23
Debug ARP	10-24
Debug Broadcast	10-25
Debug DECnet-Connects	10-27
Debug Frame-Relay	10-28
Debug Frame-Relay-Events	10-30
Debug Frame-Relay-LMI	10-31
Debug Frame-Relay-Packets	10-34
Debug IP-ICMP	10-36

Debug IP-IGRP 10-40
Debug IP-IGRP-Events 10-42
Debug IP-OSPF-Events 10-43
Debug IP-Packet 10-44
Debug IP-RIP 10-47
Debug IP-TCP 10-48
Debug LAPB 10-50
Debug LNM-Events 10-54
Debug LNM-LLC 10-56
Debug LNM-MAC 10-59
Debug Local-ACK-State 10-61
Debug Novell-Packet 10-62
Debug Novell-Routing 10-63
Debug Novell-SAP 10-64
Debug Packet 10-68
Debug RIF 10-70
Debug Serial-Interface 10-74
Debug Serial-Packet 10-83
Debug Source-Event 10-86
Debug Span 10-91
Debug TFTP 10-94
Debug Token-Ring 10-95
Debug VINES-ARP 10-98
Debug VINES-Echo 10-99
Debug VINES-Packet 10-100
Debug VINES-Routing 10-101
Debug VINES-Table 10-102
Debug XNS-Packet 10-103
Debug XNS-Routing 10-104
Debug X25 10-105
Debug X25-Events 10-109
Debug X25-VC 10-110

Appendixes

Appendix A X.25 Cause and Diagnostic Codes A-1

X.25 Cause Codes A-1

X.25 Diagnostic Codes A-3

Appendix B Technical Support Information List B-1

Gathering Information About Your Internet B-1

Appendix C Problem-Solving Checklist/Worksheet C-1

Troubleshooting Checklist C-1

Troubleshooting Worksheet C-2

Appendix D Creating Core Dumps D-1

Appendix E References and Recommended Reading

Commercially Available Publications E-1

Technical Publications and Standards E-1

Index

List of Figures

- Figure 1-1* General Problem-Solving Flow Diagram 1-4
- Figure 2-1* Initial AppleTalk Connectivity Scenario Map 2-3
- Figure 2-2* AppleTalk Zone and Network Number/Cable Range Assignments 2-6
- Figure 2-3* Show AppleTalk Interface Ethernet 6 Command Output 2-7
- Figure 2-4* Disabling AppleTalk for the Router 2-7
- Figure 2-5* Example Show AppleTalk Route 2 Command Output 2-8
- Figure 2-6* Standard Output of Show AppleTalk Global Command 2-9
- Figure 2-7* Example Show AppleTalk Neighbor Display Output 2-10
- Figure 2-8* Example Show AppleTalk Traffic Display Output 2-11
- Figure 2-9* Complete Router-R1 Final Configuration 2-13
- Figure 2-10* Initial SRB/Routing Internet Problem Environment 2-15
- Figure 2-11* Example Multiring Command Specification 2-17
- Figure 2-12* Example Output of Show RIF EXEC Command 2-18
- Figure 2-13* Example Output of Show ARP EXEC Command 2-18
- Figure 2-14* Relevant Router-Corp Final Configuration 2-19
- Figure 2-15* Relevant Router-Far Final Configuration 2-20
- Figure 2-16* Initial IBM Internet Problem Environment 2-22
- Figure 2-17* Sniffer Output Showing SRB-Capable End System Source Address 2-24
- Figure 2-18* Sniffer Output Showing End System Packet with RIF 2-25
- Figure 2-19* Reconfigured IBM Internet Environment 2-28
- Figure 2-20* Relevant Router-1 Final Configuration Listing 2-29
- Figure 2-21* Relevant Router-3 Final Configuration Listing 2-30
- Figure 2-22* Relevant Router-4 Final Configuration Listing 2-31
- Figure 2-23* Relevant Router-5 Final Configuration Listing 2-32
- Figure 2-24* Initial Novell IPX Connectivity Scenario Map 2-33
- Figure 2-25* IPX Connectivity Map Showing Revised Network Number 2-37
- Figure 2-26* All Nets Helper Address Specification Illustration 2-40

Figure 2-27 Novell IPX Connectivity Scenario Map with Backdoor Bridge Shown 2-42

Figure 2-28 Relevant Router-D Configuration Commands 2-43

Figure 2-29 Relevant Router-M Configuration Commands 2-43

Figure 2-30 TCP/IP Internetwork Connectivity Scenario Map 2-45

Figure 2-31 Example RIP-to-IGRP Route Redistribution Configuration 2-47

Figure 2-32 Access Control Additions to Router-Eng Configuration 2-47

Figure 2-33 Standard Access Control for Router-Eng Configuration 2-48

Figure 2-34 Extended Access Control for Router-Eng Configuration 2-48

Figure 2-35 Complete Example Configuration for Router-Eng 2-50

Figure 2-36 X.25 WAN Connectivity Scenario Map 2-52

Figure 2-37 Display Output of Show Version Command 2-53

Figure 2-38 Example Output of Show Controllers MCI Command 2-54

Figure 2-39 Show Interface Serial Command Indicating Bad Hardware 2-54

Figure 2-40 Show Interfaces Output Indicating Link Is Up After Cable Swap 2-55

Figure 2-41 Show Interface Ethernet Output for Operational Interface 2-56

Figure 2-42 Successful First Ping Communication from Router-New to Target Host 2-56

Figure 2-43 Transmission of Second Ping to Target Host After Clearing ARP Cache 2-57

Figure 2-44 Example Output of Show Arp Command Before Ping 2-57

Figure 2-45 Example Output of Show Arp Command After Ping 2-57

Figure 2-46 Output of Debug X.25 Events Command 2-58

Figure 2-47 Complete Configuration Showing Changes Needed to Pass Traffic 2-60

Figure 3-1 Example Show AppleTalk Interface Display Illustrating Port Mismatch 3-4

Figure 4-1 Checking IBM Serial Link to Router with Breakout Box 4-14

Figure 5-1 Basic Helper Address Network 5-13

Figure 5-2 Single Serial Interconnection Helper Address Network 5-14

Figure 5-3 All Nets Multiple Serial Line Helper Address Specification 5-15

Figure 5-4 Directed Broadcast Helper Address Specification 5-16

Figure 5-5 Reverse Broadcast Helper Address Network 5-17

Figure 5-6 Novell Helper Address Handling with Parallel Routers 5-19

Figure 5-7 Example Network Diagram Illustrating Novell-to-X.25 Mapping 5-22

Figure 5-8 Example Network Diagram Illustrating Novell-to-Frame Relay Mapping 5-23

Figure 6-1 Host-A Cannot Communicate with Host-B over Routers 6-4

-
- Figure 6-2* Problem Parallel Path Configuration Example 6-9
- Figure 7-1* Display Output for HDLC Version of Show Interfaces 7-3
- Figure 7-2* Show Interfaces Serial Diagnostic Field Locations 7-6
- Figure 7-3* Display Output for X.25 Version of Show Interfaces 7-12
- Figure 7-4* Example Display Output of Show Controllers MCI Command 7-13
- Figure 7-5* Extended Ping Specification Menu 7-17
- Figure 8-1* Upgrade from Dial-Up Link to 9600-Baud Connection 8-3
- Figure 8-2* Novell IPX Interconnection Converted from Bridging to Routing 8-5
- Figure 8-3* Novell IPX Interconnection over Router Joining 16-Mbps Rings 8-7
- Figure 8-4* Novell IPX Router Joining Ethernet and Token Ring 8-9
- Figure 8-5* Alternative Solutions to Ethernet Backbone Bottleneck 8-11
- Figure 8-6* Router Joining Novell IPX Networks over Parallel T1 Lines 8-13
- Figure 8-7* Router Joining Novell IPX Networks over Uneven Parallel Lines 8-15
- Figure 8-8* Dual 56-Kbps Serial Link TCP/IP Internet Scenario Map 8-17
- Figure 8-9* Display Output of Show Interfaces Command 8-18
- Figure 8-10* Example Ping Command Specification and Output 8-19
- Figure 8-11* Configuration Showing Priority Queuing Specification 8-20
- Figure 8-12* 56-Kbps Point-to-Point Performance Problem Scenario Map 8-22
- Figure 8-13* Display Output of Show Interfaces Command 8-23
- Figure 8-14* Show Buffers Command Output 8-24
- Figure 8-15* Complete Configuration Showing Changes Needed 8-26
- Figure 9-1* Load Balancing Problem Map 9-10
- Figure 10-1* Example Debug Broadcast Output 10-3
- Figure 10-2* Example Debug Apple-ARP Output 10-9
- Figure 10-3* Example Debug Apple-Errors Output 10-10
- Figure 10-4* Example Debug Apple-Events Output with Discovery Mode State Changes 10-12
- Figure 10-5* Example Debug Apple-Events Output Showing Seed Coming Up by Itself 10-14
- Figure 10-6* Example Debug Apple-Events Output Showing NonSeed with No Seed 10-15
- Figure 10-7* Example Debug Apple-Events Output Showing Compatibility Conflict 10-15
- Figure 10-8* Example Debug Apple-NBP Output 10-16
- Figure 10-9* Example Debug Apple-Packet Output 10-19
- Figure 10-10* Example Debug Apple-Routing Output 10-21

Figure 10-11 Example Debug Apple-ZIP Output 10-23

Figure 10-12 Example Debug ARP Output 10-24

Figure 10-13 Example Debug Broadcast Output 10-25

Figure 10-14 Example Debug DECnet-Connect Output 10-27

Figure 10-15 Example Debug Frame-Relay-Packets Output 10-28

Figure 10-16 Example Debug Frame-Relay-Events Output 10-30

Figure 10-17 Example Debug Frame-Relay-LMI Output 10-31

Figure 10-18 Example Debug Frame-Relay-Packets Output 10-34

Figure 10-19 Example Debug IP-ICMP Output 10-36

Figure 10-20 Example Debug IP-IGRP Output 10-40

Figure 10-21 Example Debug IP-IGRP Output 10-42

Figure 10-22 Example Debug IP-OSPF-Events Output 10-43

Figure 10-23 Example Debug IP-Packet Output 10-44

Figure 10-24 Example Debug IP-RIP Output 10-47

Figure 10-25 Example Debug IP-TCP Output 10-48

Figure 10-26 Example Debug LAPB Output—Part 1 10-50

Figure 10-27 Example Debug LMN-Events Output 10-54

Figure 10-28 Example Debug LMN-LLC Output 10-56

Figure 10-29 Example Debug LNM-MAC Output 10-59

Figure 10-30 Example Debug Local-ACK-State Output 10-61

Figure 10-31 Example Debug Novell-Packet Output 10-62

Figure 10-32 Example Debug Novell-Routing Output 10-63

Figure 10-33 Example Debug Novell-SAP Output 10-64

Figure 10-34 Example Debug Packet Output 10-68

Figure 10-35 Example Debug RIF Output 10-70

Figure 10-36 Example Debug Serial-Interface Output for HDLC 10-76

Figure 10-37 Example Debug Serial-Interface Output for PPP 10-81

Figure 10-38 Example Debug Serial-Interface Output When CHAP Is Enabled on a PPP Interface 10-81

Figure 10-39 Example Debug Serial-Packet Output for PPP 10-83

Figure 10-40 Example Debug Serial-Packet Output for SMDS 10-85

Figure 10-41 Example Debug Source-Event Output 10-86

Figure 10-42 Example Debug Span Output 10-91

-
- Figure 10-43* Example Debug Span Output 10-92
- Figure 10-44* Example Debug TFTP Output 10-94
- Figure 10-45* Example Debug Token-Ring Output 10-95
- Figure 10-46* Example Debug VINES-ARP Output 10-98
- Figure 10-47* Example Debug VINES-Echo Output 10-99
- Figure 10-48* Example Debug VINES-Packet Output 10-100
- Figure 10-49* Example Debug VINES-Routing Output 10-101
- Figure 10-50* Example Debug VINES-Table Output 10-102
- Figure 10-51* Example Debug XNS-Packet Output. 10-103
- Figure 10-52* Example Debug XNS-Routing Output 10-104
- Figure 10-53* Example Debug X25 Output 10-105
- Figure 10-54* Example Debug X25-Events Output 10-109
- Figure 10-55* Example Debug X25-VC Output 10-110

List of Tables

- Table 1-1* Power-up Problem Symptoms and Possible Causes 1-12
- Table 1-2* Failure Symptoms by Card or Product Type. 1-14
- Table 1-3* Suggested Actions for Ethernet Problems 1-20
- Table 1-4* Suggested Actions for Serial Line Problems 1-20
- Table 1-5* Suggested Actions for Token Ring Problems 1-20
- Table 1-6* Suggested Actions for FDDI Problems 1-21
- Table 3-1* Comparison of Phase 1 and Phase 2 NBP Packet Types 3-5
- Table 3-2* AppleTalk Problem Prevention Suggestions 3-7
- Table 3-3* Causes and Actions for Blocked Access to Offnet Resources 3-13
- Table 3-4* Causes and Actions for Missing Services 3-14
- Table 3-5* Causes and Actions for Interface Failing to Start AppleTalk 3-16
- Table 3-6* Causes and Actions for Zones Not Appearing 3-18
- Table 3-7* Causes and Actions for Intermittent AppleTalk Service Loss 3-20
- Table 3-8* Causes and Actions for Blocked Service Access 3-22
- Table 3-9* Causes and Actions for Zone List Constantly Changing 3-23
- Table 3-10* Causes and Actions for Services Being Dropped 3-24
- Table 3-11* Causes and Actions for Stuck Port Problem 3-25
- Table 3-12* Causes and Actions for Zones with Missing Network Numbers. 3-26
- Table 4-1* Causes and Actions for Blocked Routing in SRB Environments 4-4
- Table 4-2* Causes and Actions for Routing Failures in SRB Networks 4-5
- Table 4-3* Causes and Actions for Blocked SRB Traffic 4-6
- Table 4-4* Causes and Actions for Remote SRB Communication Problems 4-8
- Table 4-5* Causes and Actions for Intermittent Connectivity over Remote SRB 4-9
- Table 4-6* Causes and Actions for Traffic Stoppages over a Translational Bridge 4-10
- Table 4-7* Causes and Actions for SRT Communication Problems 4-12
- Table 4-8* Causes and Actions for SDLLC Communication Problems 4-13
- Table 4-9* Key RS-232 Signaling Requirements for Router to IBM FEP Connection 4-14

<i>Table 4-10</i>	Causes and Actions for Intermittent SDLC Connectivity	4-16
<i>Table 4-11</i>	Causes and Actions for Router Not Connecting to Ring	4-17
<i>Table 4-12</i>	Causes and Actions for Blocked Communication with SDLC Device	4-19
<i>Table 4-13</i>	Causes and Actions when SDLC Sessions Fail over STUN	4-20
<i>Table 4-14</i>	Causes and Actions for Blocked NetBIOS Communication	4-22
<i>Table 4-15</i>	Causes and Actions for Problems Linking Router via LNM	4-23
<i>Table 5-1</i>	Causes and Actions for Blocked NetWare Connectivity over Router	5-4
<i>Table 5-2</i>	Causes and Actions for SAP Updates Not Being Broadcast	5-9
<i>Table 5-3</i>	Causes and Actions for Blocked NetBIOS Traffic	5-11
<i>Table 5-4</i>	Causes and Actions for Blocked Novell Traffic over PSNs	5-21
<i>Table 6-1</i>	Causes and Actions for Blocked Access to Remote Hosts	6-4
<i>Table 6-2</i>	Causes and Actions for Unreachable Network Problems	6-6
<i>Table 6-3</i>	Causes and Actions for Selectively Blocked Host Access	6-7
<i>Table 6-4</i>	Causes and Actions for Selective Service Availability	6-8
<i>Table 6-5</i>	Causes and Actions for an Inadvertently Blocked Parallel Path	6-10
<i>Table 6-6</i>	Causes and Actions for Duplicate Routing Updates and Packets	6-11
<i>Table 6-7</i>	Causes and Actions for Some Protocols Not Being Routed	6-12
<i>Table 6-8</i>	Causes and Actions for Unreachable Hosts on Same Major Network	6-13
<i>Table 6-9</i>	Comparison of Host and Router Subnet Mask Effects	6-14
<i>Table 6-10</i>	Causes and Actions for Route Redistribution Problems	6-15
<i>Table 7-1</i>	Show Interfaces Serial Status Line Problem States	7-4
<i>Table 7-2</i>	Meaning of Key Input Errors for Serial Line Troubleshooting	7-7
<i>Table 7-3</i>	Sources of and Suggested Remedies for Clocking Problems	7-19
<i>Table 7-4</i>	Range of Packet Sizes Held in System Buffers	7-21
<i>Table 7-5</i>	Causes and Actions for WAN Intermittent Connectivity	7-26
<i>Table 7-6</i>	Causes and Actions for Load-Related WAN Problems	7-27
<i>Table 7-7</i>	Causes and Actions for Time-of-Day WAN Problems	7-28
<i>Table 7-8</i>	Causes and Actions for “Sudden Death” WAN Problems	7-29
<i>Table 7-9</i>	Causes and Actions for New Router Problems (Serial HDLC)	7-30
<i>Table 7-10</i>	Causes and Actions for New Router Problems (X.25)	7-31
<i>Table 7-11</i>	Causes and Actions for New Router Problems (Frame Relay)	7-33
<i>Table 7-12</i>	Causes and Actions for New Router Problems (SMDS)	7-35

<i>Table 7-13</i>	Causes and Actions for Selective Connectivity Problems	7-37
<i>Table 9-1</i>	Causes and Actions for Poor AppleTalk Internet Performance	9-3
<i>Table 9-2</i>	Causes and Actions for Load-Related RSRB Performance Problems	9-5
<i>Table 9-3</i>	Causes and Actions for Novell Performance Problems in LAN Internet	9-6
<i>Table 9-4</i>	Causes and Actions for Novell Performance Problems in WAN	9-7
<i>Table 9-5</i>	Causes and Actions for Slow Performance in TCP/IP Internets.	9-8
<i>Table 9-6</i>	Causes and Actions for Poor Performance Because of Blocked Paths	9-9
<i>Table 9-7</i>	Causes and Actions for Load-Related WAN Performance Problems	9-11
<i>Table 9-8</i>	Causes and Actions for WAN Performance-Related Loss of Connections	9-13
<i>Table 10-1</i>	Logging Message Keywords and Levels	10-4
<i>Table 10-2</i>	Debug Apple-NBP Field Descriptions—Part 1	10-17
<i>Table 10-3</i>	Debug Apple-NBP Field Descriptions—Part 2	10-17
<i>Table 10-4</i>	Debug Apple-Packet Field Descriptions—Part 1	10-19
<i>Table 10-5</i>	Debug Apple-Packet Field Descriptions—Part 2	10-20
<i>Table 10-6</i>	Debug Apple-Routing Field Descriptions—Part 1	10-21
<i>Table 10-7</i>	Debug Apple-Routing Field Descriptions—Part 2	10-22
<i>Table 10-8</i>	Debug Broadcast Field Descriptions	10-25
<i>Table 10-9</i>	Debug DECnet-Connects Field Descriptions	10-27
<i>Table 10-10</i>	Debug Frame-Relay Field Descriptions	10-29
<i>Table 10-11</i>	Debug Frame-Relay-LMI Field Descriptions—Part 1	10-32
<i>Table 10-12</i>	Debug Frame-Relay-LMI Field Descriptions—Part 2	10-32
<i>Table 10-13</i>	Debug Frame-Relay-LMI Field Descriptions—Part 3	10-33
<i>Table 10-14</i>	Debug Frame-Relay-Packets Field Descriptions	10-35
<i>Table 10-15</i>	Debug IP-ICMP Field Descriptions—Part 1	10-37
<i>Table 10-16</i>	Debug IP-ICMP Field Descriptions—Part 2	10-38
<i>Table 10-17</i>	Debug IP-Packet Field Descriptions	10-45
<i>Table 10-18</i>	Security Actions	10-45
<i>Table 10-19</i>	Debug IP-TCP Field Descriptions	10-49
<i>Table 10-20</i>	Debug LAPB Field Descriptions—Part 1	10-51
<i>Table 10-21</i>	Debug LNM-LLC Field Descriptions	10-57
<i>Table 10-22</i>	Debug LNM-MAC Field Descriptions	10-60
<i>Table 10-23</i>	Debug Local-ACK-State Field Descriptions	10-61

<i>Table 10-24</i>	Debug Novell-Packet Field Descriptions	10-62
<i>Table 10-25</i>	Debug Novell-Routing Field Descriptions	10-63
<i>Table 10-26</i>	Debug Novell-SAP Field Descriptions—Part 1	10-65
<i>Table 10-27</i>	Debug Novell-SAP Field Descriptions—Part 2	10-66
<i>Table 10-28</i>	Debug Novell-SAP Field Descriptions—Part 3	10-67
<i>Table 10-29</i>	Debug Packet Field Descriptions	10-68
<i>Table 10-30</i>	Debug RIF Field Descriptions—Part 1	10-71
<i>Table 10-31</i>	Debug RIF Field Descriptions—Part 2	10-72
<i>Table 10-32</i>	Debug Serial-Interface Field Descriptions for DDR	10-75
<i>Table 10-33</i>	Debug Serial-Interface Field Descriptions for HDLC	10-77
<i>Table 10-34</i>	Debug Serial-Interface Error Messages for HDLC	10-78
<i>Table 10-35</i>	Debug Serial-Interface Field Descriptions for ISDN Basic Rate	10-79
<i>Table 10-36</i>	Debug Serial-Interface Field Descriptions for an MK5025 Device	10-80
<i>Table 10-37</i>	Debug Serial-Packet Field Descriptions for PPP	10-84
<i>Table 10-38</i>	Debug Source-Event Field Descriptions	10-86
<i>Table 10-39</i>	Debug Span Field Descriptions for an IEEE BPDU Packet	10-92
<i>Table 10-40</i>	Debug Span Field Descriptions for a DEC BPDU Packet	10-93
<i>Table 10-41</i>	Debug TFTP Field Descriptions	10-94
<i>Table 10-42</i>	Debug Token Ring Field Descriptions—Part 1	10-96
<i>Table 10-43</i>	Debug Token Ring Field Descriptions—Part 2	10-96
<i>Table 10-44</i>	Debug Token Ring Field Descriptions—Part 3	10-97
<i>Table 10-45</i>	Debug VINES-ARP Field Descriptions	10-98
<i>Table 10-46</i>	Debug VINES-Echo Field Descriptions	10-99
<i>Table 10-47</i>	Debug VINES-Packet Field Descriptions	10-100
<i>Table 10-48</i>	Debug VINES-Table Field Descriptions	10-102
<i>Table 10-49</i>	Debug XNS-Packet Field Descriptions	10-103
<i>Table 10-50</i>	Debug XNS-Routing Field Descriptions	10-104
<i>Table 10-51</i>	Debug X25 Field Descriptions	10-106
<i>Table 10-52</i>	Debug X25 PS and PR Field Descriptions	10-108

Table A-1 Cause Code Descriptions for CLEAR REQUEST Packets A-2

Table A-2 Cause Code Descriptions for RESET REQUEST Packets A-2

Table A-3 Cause Code Descriptions for RESTART Packets A-3

Table A-4 Diagnostic Field Code Descriptions A-4

About This Manual

This front matter section introduces the *Troubleshooting Internetworking Systems* audience and scope, organization, use, and conventions.

Audience and Scope

This publication addresses the network administrator or system administrator who will maintain a Cisco gateway, router, or bridge running Release 9.1 and earlier software. This release of the manual focuses on generic WAN/serial, TCP/IP, Novell IPX, IBM/SNA, and AppleTalk internets.

Readers should know how to configure a Cisco router and should be familiar with the protocols and media their routers have been configured to support. Awareness of their networking topology is also important.

Document Organization and Use

The *Troubleshooting Internetworking Systems* guide provides information about troubleshooting Cisco network servers.

Chapter 1, "Troubleshooting Overview," introduces a generic model of problem solving and provides basic information regarding troubleshooting Cisco internetworks. It is important that you read this chapter first before proceeding to other chapters of the manual.

Chapter 2, "Connectivity Problem Scenarios," presents problem-solving scenarios that focus on identifying, isolating, and solving internetworking connectivity problems. Scenarios for WAN/serial, TCP/IP, Novell/IPX, IBM/SNA, and Appletalk internets are included. Each of these describes the symptoms of the problem(s), the internetworking environment, problem cause alternatives, the process of isolating those causes, and a summary of the process.

Chapter 3, "Troubleshooting Apple Connectivity," presents protocol-related troubleshooting information for AppleTalk networks, including AppleTalk internetworking terminology, AppleTalk internetworking diagnostic tips, preventing AppleTalk configuration problems, and symptom modules. These symptom modules provide snapshots of common symptoms, possible causes, and suggested actions to resolve the problems.

Chapter 4, “Troubleshooting IBM Connectivity,” presents protocol-related troubleshooting information for IBM networks, including diagnosing IBM network and Token Ring problems and symptom modules. These symptom modules provide snapshots of common symptoms, possible causes, and suggested actions to resolve the problems.

Chapter 5, “Troubleshooting Novell Connectivity,” presents protocol-related troubleshooting information for Novell networks, including a Novell IPX Internet diagnostic overview and symptom modules. These symptom modules provide snapshots of common symptoms, possible causes, and suggested actions to resolve the problems.

Chapter 6, “Troubleshooting TCP/IP Connectivity,” presents protocol-related troubleshooting information for TCP/IP networks, including a TCP/IP Internet diagnostic overview and symptom modules. These symptom modules provide snapshots of common symptoms, possible causes, and suggested actions to resolve the problems.

Chapter 7, “Troubleshooting WAN Connectivity,” presents protocol-related troubleshooting information for IBM networks, including diagnosing WAN and serial line problems and symptom modules. These symptom modules provide snapshots of common symptoms, possible causes, and suggested actions to resolve the problems.

Chapter 8, “Performance Problem Scenarios,” presents problem-solving scenarios that focus on identifying, isolating, and solving internetworking performance problems. Scenarios for WAN/serial, TCP/IP, Novell/IPX, IBM/SNA, and Appletalk internets are included. Each of these sections describes the symptoms of the problem(s), the internetworking environment, problem cause alternatives, the process of isolating those causes, and a summary of the process.

Chapter 9, “Troubleshooting Internet Performance,” focuses on common symptoms associated with poor performance in internetworks, possible causes of those symptoms, and general suggestions for identifying, isolating, and resolving causes.

Chapter 10, “Debug Command Reference,” presents reference information on over 40 commands that you can use as tools to debug your internetwork. Descriptions of the uses of these commands, sample output displays, and explanations of these displays are included.

Appendix A, “X.25 Cause and Diagnostics Codes,” lists the codes that can appear in output from the Debug X.25, Debug X.25-Events, and Debug X.25-VC debugging commands.

Appendix B, “Technical Support Information List,” lists the information you can provide your technical support representative in order to speed up and facilitate problem resolution.

Appendix C, “Problem Solving Checklist/Worksheet,” includes a worksheet that you can use to structure your strategy for problem isolation and resolution.

Appendix D, “Creating Core Dumps,” describes the various ways that you can capture core dump information following a router crash and transmit that information to your technical support representative for further analysis.

Appendix E, “References and Recommended Reading,” lists commercially available publications that provide background information on troubleshooting internetworks and the protocols covered in this guide. It also includes a section listing other technical publications that you may find useful.

Document Conventions

The command descriptions use these conventions:

- Commands and keywords are in **boldface**.
- Filenames, directory names, and variables for which you supply values are in *italics*.
- Elements in square brackets ([]) are optional.
- Alternative but required keywords are grouped in braces ({ }) and are separated by a vertical bar (|).
- A string is defined as a nonquoted set of characters. For example, when setting up a community string for SNMP to “public”, do not use quotes around the string or the string will be set to “public”.

The samples use these conventions:

- Terminal sessions are printed in a `screen font`.
- Information you enter is in a **boldface screen font**.
- Nonprinting characters are shown in angle brackets (<>).
- Information the system displays is in `screen font`, with default responses in square brackets ([]).

This publication also uses the following conventions:

Note: is a special paragraph that means *reader take note*. It usually refers to helpful suggestions, the writer’s assumptions, or reference to materials not contained in this manual.



Caution: is a special paragraph that means *reader be careful*. It means that you are capable of doing something that might result in equipment damage, or worse, that you might have to take something apart and start over again.

Related Documentation

Following is a list of related publications shipped with the router product:

- *Router Products Configuration and Reference*
- *Internetworking Terms and Acronyms*
- *System Error Messages*
- *Router Products Getting Started Guide*

- Configuration notes for your router product, if applicable
- Hardware installation and maintenance publication for your router product

To order these publications or additional copies of *Troubleshooting Internetworking Systems*, contact your sales representative. (The Customer Order Number for each manual is located at the bottom of the title page.) Customer Service can provide you with the name of your sales representative if necessary.

Phone: 1-800-553-NETS (6387) or (415) 326-1941

E-mail: customer-service@cisco.com

Service and Support

Cisco Systems provides a full range of support services to ensure that you get maximum network uptime with low life-cycle equipment cost. This section contains instructions for contacting Customer Service and for obtaining assistance through the Technical Assistance Center (TAC). It also contains warranty and service information.

Warranty Information

All Cisco Systems products are covered under a limited factory warranty. This warranty covers defects in the hardware, software, or firmware. Refer to the Cisco Systems *Customer Services Product Guide* for more information on Cisco's warranty policy, or contact Customer Service at 1-800-553-NETS or (415) 326-1941.

Note: Warranty and other service agreements may differ for international customers. Contact your closest Cisco regional representative for more information.

Maintenance Agreements

Cisco Systems offers a Comprehensive Hardware Maintenance Agreement throughout North America that includes on-site remedial services, software support, a 24-hour emergency hot line, overnight parts replacement, and an escalation procedure. Cisco also offers software, maintenance, and advanced replacement services under a SMARTnet agreement for customers who desire those services. Noncontract maintenance services are provided at current time-and-materials rates. For more information, contact Customer Service at 1-800-553-NETS or (415) 326-1941.

Customer Support

Cisco's maintenance strategy is based upon customer-initiated service requests to the Cisco Systems Technical Assistance Center (TAC). The TAC coordinates all customer services, including hardware and software telephone technical support, onsite service requirements, and module exchange and repair.

The TAC is available Monday through Friday from 6:00 a.m. to 6:00 p.m. Pacific Coast time (excluding company holidays) at the numbers that follow. If you must return your Cisco equipment for repair or replacement, contact the TAC or a Cisco regional representative for more information.

Hardware and software support specialists who help diagnose and solve customer problems will be able to isolate and solve your problem much faster if you are prepared with the information they need (see the TAC escalation procedures page shipped with this product). When you call the TAC, have the following information ready:

- Chassis serial number
- Maintenance contract number
- Software version and hardware configuration

You can display your software version level and your hardware configuration by using the **show version** command.

Technical Assistance (TAC):

1-800-553-2447	Fax: (415) 903-8787
(415) 688-8209	E-mail: tac@cisco.com

Sales, Orders, Questions, and Comments:

1-800-553-NETS (6387)	Fax: (415) 903-8080
(415) 903-7208	E-mail: csrep@cisco.com

Obtaining Additional Information

This section describes how to obtain additional Cisco publications and includes tips for obtaining books, standards, and other information about networks and data communications that might be helpful while using Cisco products.

Ordering Additional Cisco Publications

To order these publications or additional copies of *Troubleshooting Internetworking Systems*, contact your sales representative. (The Customer Order Number for each manual is located at the bottom of the title page.) Customer Service can provide you with the name of your sales representative if necessary.

1-800-553-NETS (6387)

(415) 326-1941

E-mail: customer-service@cisco.com

Obtaining Cisco Technical Information Electronically

Cisco provides a directory of documents that you can access electronically using File Transfer Protocol (FTP). The directory includes such publications as product release notes, descriptions of Management Information Bases (MIBs), commonly used Requests for Comments (RFCs), and technical notes. The directory does not include electronic versions of Cisco technical manuals.

To obtain these technical documents, proceed as follows:

Step 1: At your server prompt, use the **ftp** command to connect to address *ftp.cisco.com*.

```
% ftp ftp.cisco.com
```

When you connect to the directory, you are greeted with an informational banner:

```
Connected to dirt.cisco.com.  
220 dirt FTP server (Version 5.51.28 Mon Jan 13 17:51:58 PST 1992)  
ready.
```

This is followed by a login prompt.

Step 2: Enter the word **anonymous** as your login name:

```
Name (ftp.cisco.com:cindy): anonymous
```

The system responds with this message:

```
331 Guest login ok, send ident as password.  
Password:
```

Step 3: Enter your login name at the Password: prompt. The following message and ftp> prompt appear:

```
230 Guest login ok, access restrictions apply.  
ftp>
```

Step 4: To obtain a list of available files, enter **get README** at the ftp> prompt:

```
ftp> get README  
200 PORT command successful.  
150 Opening ASCII mode data connection for README (10093 bytes).  
226 Transfer complete.  
local: README remote: README  
10307 bytes received in 0.17 seconds (59 Kbytes/s)
```

Step 5: Enter the **get** command and the full filename for each file you require.

Step 6: To exit FTP, use the **quit** command.

```
ftp> quit  
221 Goodbye.
```

Note: In the FTP directory, the **ls** command does not accept wildcards; therefore, you cannot use this command to obtain a list of available files. To obtain a list of available files, you must use the README file.

Obtaining Information from Other Sources

This section describes how to obtain RFCs and technical standards.

For a list of relevant publications from other sources, see Appendix E, “References and Recommended Reading.”

Obtaining RFCs

Information about the Internet suite of protocols is contained in documents called *Requests for Comments*, or *RFCs*. These documents are maintained by Government Systems, Inc. (GSI). You can request copies by contacting GSI directly, or you can use the TCP/IP File Transfer Protocol (FTP) to obtain an electronic copy.

Contacting GSI

You can contact GSI through mail, by telephone, or through electronic mail.

Government Systems, Incorporated
Attn: Network Information Center
14200 Park Meadow Drive, Suite 200
Chantilly, Virginia 22021

1-800-365-3642
(703) 802-4535
(703) 802-8376 (FAX)

NIC@NIC.DDN.MIL

Network address: 192.112.36.5

Root domain server: 192.112.36.4

Obtaining an Electronic Copy

To obtain an electronic copy of an RFC via FTP, complete the following steps:

Step 1: At your server prompt, use the **ftp** command to connect to address *nic.ddn.mil*:

```
% ftp nic.ddn.mil
```

The following display appears, followed by a login prompt:

```
Connected to nic.ddn.mil.
```

```
220-*****Welcome to the Network Information Center*****
```

```
*****Login with username "anonymous" and password "guest"
```

```
*****You may change directories to the following:
```

ddn-news	- DDN Management Bulletins
domain	- Root Domain Zone Files
ien	- Internet Engineering Notes
iesg	- IETF Steering Group
ietf	- Internet Engineering Task Force
internet-drafts	- Internet Drafts
netinfo	- NIC Information Files
netprog	- Guest Software (ex. whois.c)
protocols	- TCP-IP & OSI Documents
rfc	- RFC Repository
scc	- DDN Security Bulletins

```
220 And more.
```

Step 2: At the login prompt, enter the word **anonymous** as your login name:

```
Name (nic.ddn.mil:cindy): anonymous
```

The NIC responds with this message:

```
331 Guest login ok, send "guest" as password.
```

```
Password:
```

Step 3: Enter the word **guest** at the Password: prompt. The following message and ftp> prompt appear:

```
230 Guest login ok, access restrictions apply.
```

```
ftp>
```

Step 4: Use the **cd** command to change directories. The following example illustrates how to change the RFC directory and obtain RFC 1158:

```
ftp> cd rfc  
250 CWD command successful.  
ftp> get rfc1158.txt
```

Step 5: To exit the FTP facility, enter the **quit** command at the ftp> prompt.

Obtaining Technical Standards

Following are additional sources for technical standards:

- Omnicom, 1-800-OMNICOM
- Global Engineering Documents, 2805 McGraw Ave., Irvine, CA 92714
1-800-854-7179
- American National Standards Institute, 1430 Broadway, New York, NY 10018
(212) 642-4932 or (212) 302-1286

Chapter 1

Troubleshooting Overview

1

Focus on Symptoms, Causes, and Actions 1-1

What This Guide Is Not 1-2

Using This Publication 1-3

General Problem-Solving Model 1-3

Problem-Solving Model Components 1-3

Symptom Modules 1-5

Troubleshooting Scenarios 1-6

Using This Publication to Troubleshoot Specific Symptoms 1-6

Using This Publication as a Tutorial 1-6

Using This Publication with Other Cisco Publications 1-7

Using Cisco Diagnostic Tools 1-8

Using Show Commands 1-8

Using Debug Commands 1-9

Using Ping and Trace Commands 1-9

Using Core Dumps 1-9

Diagnosing Cisco Hardware 1-10

Physically Inspecting Your System 1-10

Applying Power and Evaluating the System 1-11

Testing and Verifying Operation 1-13

Using CiscoWorks to Troubleshoot Your Internet 1-17

Using CiscoWorks to Troubleshoot Connectivity Problems 1-17

Using CiscoWorks to Troubleshoot Performance Problems 1-18

Using Third-Party Troubleshooting Tools 1-19

Troubleshooting Media Problems 1-20

Chapter 1

Troubleshooting Overview

1

Internetworks come in a variety of topologies and levels of complexity—from single-protocol, point-to-point links connecting cross-town campuses to highly meshed, large-scale WANs traversing multiple time zones and international boundaries. The overall trend is toward increasingly complex environments, involving multiple media, multiple protocols, and sometimes interconnection to “unknown” networks. As a result, the potential for connectivity and performance problems in internets is often high, even when all elements of an environment appear to be fully operational. The objective of this publication is to help you identify potential problem sources in your internet and then to systematically resolve problems that arise.

Note: In this release of *Troubleshooting Internetworking Systems*, coverage focuses on AppleTalk, IBM SNA, Novell IPX, TCP/IP, and WAN/serial internets. Subsequent releases will cover additional protocols and technologies.

Focus on Symptoms, Causes, and Actions

Failures in internets are characterized by certain *symptoms* (such as clients being unable to access specific servers). Each symptom can be diagnosed based on *problems* or *causes* using specific troubleshooting tools. Once identified, each cause can be remedied by implementing a series of *actions*.

Use this manual as a starting point to develop a problem-solving process for your internetwork. This publication aims to integrate the process of symptom definition, problem identification, and action implementation into an overall troubleshooting model. It illustrates how problems can be detected and diagnosed within the context of case environments.

What This Guide Is Not

With these broad objectives stated, it is equally important to outline topics that are beyond this publication's scope.

- This publication is *not* intended to be the last word in troubleshooting. You will not find every "corner case" (or obscure anomaly) and subtle protocol problem. Instead, *Troubleshooting Internetworking Systems* is a roadmap that illustrates the *common* pitfalls and problems most frequently encountered by internetwork administrators.
- *Troubleshooting Internetworking Systems* is *not* a maintenance and repair guide; nor is it a reference guide. Refer to your hardware installation and maintenance publication for additional details regarding maintenance of Cisco hardware. Refer to the *Router Products Configuration and Reference* publication for configuration command details.

This publication recommends actions for resolving a spectrum of common internetworking problems. In general, it assumes that routers are operational. However, several brief tables provided later in this chapter summarize typical router hardware problems.

- Finally, *Troubleshooting Internetworking Systems* is not a *network* troubleshooting publication. Although suggestions are provided in this chapter about troubleshooting certain media (including Ethernet, FDDI, serial, and Token Ring), the focus of the publication is not on troubleshooting media, per se. Several commercially available publications provide this information. One is Mark Miller's *LAN Troubleshooting Handbook*. Appendix E, "References and Recommended Readings," suggests some others.

What, then, does that leave? The discussions that follow outline how you *can* use this publication to resolve common internetworking problems.

The remainder of this overview addresses the following topics:

- Using this publication
- Using Cisco diagnostic tools
- Diagnosing Cisco hardware
- Using CiscoWorks to troubleshoot your internet
- Using third-party troubleshooting tools
- Troubleshooting media problems

Using This Publication

Troubleshooting Internetworking Systems focuses on identifying failure symptoms and their associated causes, detecting and isolating those causes, and then resolving problems through specific actions. The symptom discussions and scenarios provided concentrate on issues pertaining to *router* configuration and the interoperation of nodes within a multivendor internetwork.

Within this context, use *Troubleshooting Internetworking Systems* as a guide to:

- Identify possible problem causes when your internet is down or slow
- Get direction about resolving problems
- See what kinds of problems have been encountered and resolved in the past
- Avoid falling into the same traps
- Develop your own processes for troubleshooting

To support these activities, this guide uses three key organizational elements (defined in the discussions that follow):

- General problem-solving model
- Symptom tables
- Troubleshooting scenarios

In addition, this overview provides guidelines for the following tasks:

- Using this guide to troubleshoot problems
- Using this guide as a tutorial

General Problem-Solving Model

Before embarking on your troubleshooting effort, be sure to have a *plan* in place to identify prospective problems, isolate the likely causes of those problems, and then systematically eliminate each potential cause.

The problem-solving model that follows is not a rigid “cookbook” for solving internetworking problems. It is a foundation from which you can build problem-solving plans to suit your particular environment.

Problem-Solving Model Components

Figure 1-1 illustrates process flow for the general problem-solving model described in the steps that follow.

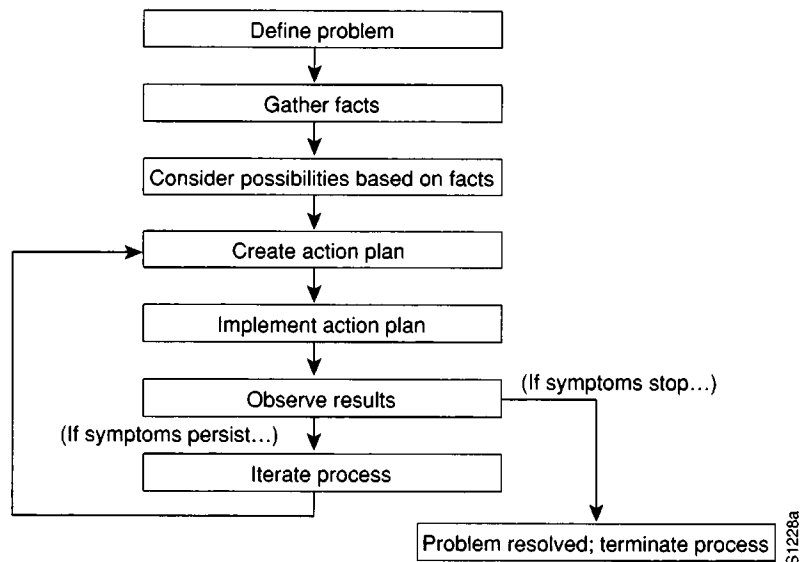


Figure 1-1 General Problem-Solving Flow Diagram

The following steps detail the problem-solving process outlined in Figure 1-1:

Step 1: Define problems in terms of a set of *symptoms* and associated *causes*.

Make a clear problem statement. You must recognize and define the problem/failure mode by identifying any associated general symptoms and then identifying the possible kinds of problems that result in the listed symptoms.

For example, certain hosts might not be responding to service requests from certain clients (a symptom). Possible causes include a misconfigured host, bad interface cards, or missing router commands.

Step 2: Gather facts.

Once your symptoms are listed and possible causes identified, collect facts. Fact gathering might involve obtaining network analyzer traces, serial line traces, stack dumps, core dumps, and output from a variety of **show** and **debug** commands. The definition of the problem will point to a more specific set of data to gather.

Step 3: Consider possibilities based on facts.

Armed with a working knowledge of the product, you should be able to eliminate entire classes of problems associated with system software and hardware. This way, you can narrow the scope of interest to only those portions of the product, media, or host problems that are relevant to the specific problem or failure mode.

Step 4: Create an action plan.

The action plan should be based on the set of possibilities you just derived. Your action plan must limit manipulation to *one* variable at a time. This approach allows you to reproduce a given solution to a specific problem. If you alter more than one variable simultaneously, you might solve the problem, but identifying the specific change that eliminated the symptom becomes more difficult.

Step 5: Implement action plan.

This phase consists of executing the action plan you just created. It is important to be very specific in creating the action plan (that is, identify a specific set of steps and then carefully implement each step).

Step 6: Observe results of each action.

After having manipulated a variable in an attempt to find a solution to a problem, be sure to gather results based on this action plan (obtain relevant traces, capture **debug** command data, examine output of **show** commands, etc.). This data can be used to fine-tune the action plan until the proper solution is achieved. It is during this phase that you must determine whether the problem has been resolved. This is the exit point of the iterative loop shown in Figure 1-1.

Step 7: Narrow possibilities based on results.

In order to reach a point where you can exit this problem/solution loop, you must strive to make continuous progress toward a smaller set of possibilities, until you are left with one.

Step 8: Iteratively apply problem-solving process.

After narrowing your possibility list, repeat the process, starting with a new action plan based on a new (possibly shorter or longer) list of possibilities. Continue the process until a solution is found. Problem resolution can consist of several modifications to hosts, routers, or media.

Note: If you exhaust all the common causes and actions (either those suggested here or ones that you have identified for your environment), your last recourse is to contact your router technical support representative. Appendix B, “Technical Support Information List,” outlines information needed by technical support representatives to troubleshoot internet-working problems. One objective of this publication is to help you develop your own processes for gathering data, resolving problems, and preventing problems from recurring (with a minimum of downtime and external intervention).

Symptom Modules

The *symptom modules* in this publication are *not* comprehensive case studies, but instead are brief snapshots of likely problems associated with a specific symptom. Use them as tools for compiling lists of candidate problems (by symptom). The connectivity and performance chapters (such as Chapter 3, “Troubleshooting Apple Connectivity”) are organized around the symptom modules. These chapters are not meant to be read from beginning to end; rather, specific information in these symptom-oriented chapters is intended to be *found* as needed.

Each symptom module includes a brief summary statement and a table listing possible causes. A series of suggested actions is provided for each listed cause to help you determine whether the specific cause is actually the source of the symptom and then to resolve the problem.

Troubleshooting Scenarios

The *troubleshooting scenarios* combine the problems and actions presented in symptom modules with the methods outlined in the “General Problem-Solving Model” within a context of integrated *case studies*.

Each scenario outlines a set of “observed” symptoms, an internetworking environment, and a list of likely problems for each symptom. Scenarios focus on the process of problem diagnosis (discovery), isolation, and resolution. Not all symptoms discussed in this publication are explored in the scenarios. Instead, selected multiple symptoms are addressed per scenario. An effort has been made to pick common, realistic problems.

Using This Publication to Troubleshoot Specific Symptoms

When using this publication to troubleshoot your internet, follow these general steps:

- Step 1:** Identify symptoms being experienced on your internetwork.
- Step 2:** Eliminate Cisco hardware as a possible problem (refer to “Diagnosing Cisco Hardware”) by either fixing any hardware problems or ruling out Cisco hardware as a possible cause.
- Step 3:** Scan the *symptom modules* provided in the various chapters addressing the technologies or protocols used in your internet to identify similar symptoms.
- Step 4:** Within identified symptom modules, evaluate problems listed in the associated “Possible Causes and Suggested Actions” section.
- Step 5:** Systematically and iteratively apply actions for each suspect problem until all symptoms are eliminated or the possible cause list is exhausted.

Using This Publication as a Tutorial

When using this guide as a tutorial, associated activities are a little less structured than when using it to troubleshoot a specific problem. Nonetheless, you can think of the learning process as a series of steps, as follows:

- Step 1:** Review “General Problem-Solving Model” earlier in this chapter to see how Cisco recommends approaching the troubleshooting process.
- Step 2:** Read through the *scenarios* presented in Chapter 2, “Connectivity Problem Scenarios,” and Chapter 8, “Performance. Problem Scenarios.”
- Step 3:** Characterize similarities or differences between these scenarios and your own internetworking environment.
- Step 4:** Review the symptom modules associated with protocols or technologies implemented in your internet.
- Step 5:** Develop a list of possible symptoms and problems that you encounter in your internet. Be as specific as possible. Keep this list on hand in a troubleshooting binder.
- Step 6:** When similar symptoms occur, use this list to start the troubleshooting process. Remember to modify your problem-solving procedures as you find subtleties

associated with your implementation. The key to developing an effective response to problems in your environment is being able to identify the causes of those problems and then implement an action plan. Whatever you can do to preempt time spent in diagnosis will pay off in terms of reducing downtime.

Step 7: Periodically revisit this process to accommodate changes and additions to your internet.

Using This Publication with Other Cisco Publications

Troubleshooting Internetworking Systems is one of several Cisco information resources that are essential for building Cisco-based internetworking environments. These resources include the following:

- *Getting Started Guides*—The getting started guides provide crucial information for first-time startup. Information includes a step-by-step initialization process. Separate manuals are provided for routers, communication servers, and protocol translators.
- *Hardware Installation and Maintenance*—The hardware installation and maintenance manuals are essential when bringing a new router on line. Information includes product overview, preinstallation information, installing the hardware (cabling, rack-mounting), troubleshooting for initial hardware configuration, user-configurable jumpers and switch settings, cabling instructions, and LED functional definitions.
- *Router Products Configuration and Reference*—The configuration and reference guide is the key system software reference publication for all information about configuring and monitoring your router. Information includes step-oriented task lists and complete reference material for configuration and system monitoring commands.
- *System Error Messages*—The error messages handbook lists and describes system error messages for routers, communication servers, and protocol translators. Not all messages indicate problems with a system. Some are informational, while others can help diagnose media, hardware, and software problems.
- *Release Notes*—Generally, a release note is the first place to look when configuring a new system. Information includes a summary of new software features, a listing and description of known bugs, descriptions of microcode revisions, a hardware/software compatibility matrix, hardware caveats, and a summary of new hardware features.
- *Configuration Notes*—Configuration notes are required reading for performing any upgrade or other change. Information includes installation and configuration instructions for spare or replacement and upgrade parts (such as cards, appliques, system software, and microcode).
- *CiscoWorks User Guide*—This guide provides detailed information about remote management of Cisco internetworking systems using the CiscoWorks network management application package.
- *Customer Information Online (CIO)*—Cisco's online information resource provides application information and current bug descriptions. This service is provided by Cisco's Customer Engineering (CE) organization. Contact your technical support representative for more information about accessing this database.

Using Cisco Diagnostic Tools

The following tools are universally applicable when gathering information to troubleshoot problems in Cisco internetworks:

- **show** EXEC commands
- **debug** diagnostic EXEC commands.
- **ping** (Echo Request/Echo Reply) and **trace** diagnostic tests
- **exception dump** and **write core** configuration commands

The discussions that follow summarize using these tools. Chapter 10, “Debug Command Reference,” defines the **debug** commands for protocols and technologies discussed in this publication, and Appendix D, “Creating Core Dumps,” describes the **exception dump** and **write core** commands.

The *Router Products Configuration and Reference* publication details the **show**, **ping**, and **trace** commands.

Using Show Commands

The router’s **show** commands are among your most important tools for understanding the status of a router, detecting neighboring routers, monitoring the network in general, and isolating problems in the internet.

These commands are essential in almost any troubleshooting and monitoring situation. Use **show** commands for the following activities:

- Monitoring router behavior during initial installation
- Monitoring normal network operation
- Isolating problem interfaces, nodes, media, or applications
- Determining when a network is congested
- Determining the status of servers, clients or other neighbors

For some protocols, such as Novell IPX and AppleTalk, the methodical use of **show** commands is one the most reliable ways to create a topology map of your internetwork. To create a topology map, use the **show** commands as follows:

- Step 1:** Use the appropriate **show protocol route** command (such as **show novell route**) to determine which neighbors are directly connected.
- Step 2:** Record the names and network addresses of all directly connected neighbors.
- Step 3:** Open a connection to each of these directly connected neighbors and obtain the output of the **show protocol route** command at those neighbors.
- Step 4:** Continue this process for all routers in your internet.

The resulting map reflects all paths to the routers in your internet.

Using Debug Commands

The **debug** EXEC commands can provide a wealth of information about the traffic being seen (or *not* seen) on an interface, error messages generated by nodes on the network, protocol-specific diagnostic packets, and other useful troubleshooting data. But beware! These commands often generate data that is of little use for a specific problem.

Use **debug** commands to isolate problems, not to monitor normal network operation. Do not use **debug** commands unless you are looking for specific types of traffic or problems and have narrowed your problems to a likely subset of causes.

This publication does not document *every* **debug** command that exists in the router code, but only those identified as especially useful for troubleshooting specific media and protocols.



Caution: Throughout this publication, the use of **debug** commands is suggested for obtaining information about network traffic and router status. Use these commands with great care. In general, it is recommended that these commands only be used under the direction of your router technical support representative when troubleshooting specific problems. Enabling debugging can disrupt operation of the router when internets are experiencing high load conditions. When you finish using a **debug** command, remember to disable it with its specific **undebug** command or with the **undebug all** command.

Using Ping and Trace Commands

Two of the most useful internetworking diagnostic tools are the **ping** and **trace** features. The *ping* capability provides a simple mechanism to determine whether packets are reaching a particular destination. The *trace* capability allows you to determine the specific path taken to a destination and where packets are stopping. Together, these functions may be two of the most important troubleshooting tools available. Trace is supported with TCP/IP, ISO CLNS, and Banyan VINES on the router. Ping is supported with AppleTalk, TCP/IP, ISO CLNS, Novell IPX, and Banyan VINES.

Using Core Dumps

The **exception dump** global configuration command and **write core** command are among the more obscure (although useful) diagnostic commands available in your router toolkit. When a router's system software fails, using the **exception dump** command to obtain a *core dump* is sometimes the only way to determine what happened. The **write core** command is useful if the router is malfunctioning but has not crashed.



Caution: Use these commands only in coordination with a qualified technical support representative. The resulting binary file must be directed to a specific syslog server and subsequently interpreted by qualified technical personnel. Appendix D, "Creating Core Dumps," briefly describes the process.

Diagnosing Cisco Hardware

Although this publication focuses on troubleshooting *overall* internetworking problems, the following tables provide some suggestions for diagnosing router hardware problems. Your installation and maintenance publication provides specific LED indicator information for system appliques and front panels.

This overview is *not* a step-by-step procedure. It is included as a “mental checklist” and should be used as a starting point for troubleshooting. The following discussion suggests a three-stage process:

- Physically inspecting your system
- Applying power and evaluating the system
- Testing and verifying operation

Each of these stages is discussed separately.

Physically Inspecting Your System

When initially evaluating a suspect system, keep the following three rules in mind:

- Contrast what *should be* happening with what *is* happening.
- Do not overlook the obvious.
- Do not alter *anything* before power-up; do not mask a possible failure.

At this stage, concentrate on problems that are obvious. Follow these inspection steps:

- Step 1:** Look for loose cards, cables, and appliques. Be sure to reseat any that are loose. When cards are new, sometimes a thin film of carbon or oxidation buildup prevents good contact. After reseating each board once or twice, you should achieve good contact.
- Step 2:** Remove the top of the chassis and inspect the interior. Are the wires to the power supply connected correctly? Are they burned?
- Step 3:** Look for burned cards, backplanes, and ribbon cables. Are there any visibly crimped or shorted wires or cables?
- Step 4:** Check for missing or loose parts, incorrectly connected cables, and anything that appears out of place. Does the unit need to be cleaned? Is there damage to the interior or exterior?

Note: Do not change anything before powering up the system for evaluation. Making changes can mask other problems. Do not alter anything, even if it appears to be out of place, so that you can determine the source of suspected hardware problems during subsequent evaluation.

Applying Power and Evaluating the System

Once you have inspected the system, apply power to the unit and observe its behavior. When applying power to a unit, remember the following rules:

- Do not overlook the obvious (does this seem familiar?).
- Do not jump to conclusions or make unnecessary assumptions.
- Make the symptoms explain the problem.

If you suspect a hardware problem, follow these steps to evaluate operational conditions upon power-up:

Step 1: Power up the system (when system is off line).

Step 2: Use a voltmeter to ensure that all the power supply voltages are within specifications. Refer to the configuration note for your power supply model.

Note: Configuration notes are only shipped with spares and replacement parts.

Step 3: Compare system behavior against symptoms outlined in Table 1-1.

Step 4: If a failure does not fit the examples in Table 1-1, verify that the software in the processor and the microcode in the various cards are labeled correctly, are in the appropriate order, and are compatible with the individual card revisions within the chassis. Refer to the release document provided with your system.

Step 5: If the system boots, use **show controllers {token|mci|fdi|cbus}** to ensure that the interface hardware addresses are nonzero. Hardware addresses of all zeros *will* cause problems in a network.

Note: If the system boot-up sequence requires a password, the memory card and circuitry are working correctly. If the configuration in memory does not match the hardware configuration, problems can occur. Possible problems include hung ports, uninitialized ports, ping failures, local and multibus timeout errors, and reboots.

Table 1-1 Power-up Problem Symptoms and Possible Causes

Symptoms at Power-Up	Possible Causes
No response from chassis	<ul style="list-style-type: none"> ■ Fuse blown (3000, 4000, and I-, M-, and C-chassis) ■ Bad power supply ■ Bad switch ■ Bad backplane ■ Bad breaker (AGS/AGS+)
No fan (MGS/CGS)	<ul style="list-style-type: none"> ■ Bad fan ■ Bad 12V power supply ■ Shorted or broken wires
No blower (A-type/AGS+)	<ul style="list-style-type: none"> ■ Bad blower ■ Bad breaker ■ Tripped breaker ■ Shorted or broken wires ■ Bad 110/220 capacitor
No LEDs on at boot	<ul style="list-style-type: none"> ■ Bad 5V power supply (no LEDs on card); box may boot ■ Shorted or broken wires
System will not boot	<ul style="list-style-type: none"> ■ Bad power supply ■ Miswired power supply ■ Bad/disconnected console cable (system still boots; no monitor output) ■ Bad processor card or card is poorly seated ■ Bad software ■ Bad memory board ■ Shorted wires
No cards show up in power-up message display	<ul style="list-style-type: none"> ■ Bad backplane ■ Bad processor/controller/interface card ■ Cards not seated in backplane ■ Bad power supply
Breaker trips or fuse blows	<ul style="list-style-type: none"> ■ Bad power supply ■ Bad backplane ■ Shorted wires ■ Load too large on power supply ■ No load on power supply ■ Bad breaker ■ Bad blower ■ Bad card
Constant or partial reboot	<ul style="list-style-type: none"> ■ Bad processor/controller/interface card ■ Bad backplane ■ Bad power supply ■ Bad software ■ Bad microcode

Testing and Verifying Operation

If replacing a part or card to remedy a suspected problem, remember the following rules:

- Make only one change at a time.
- Eliminate suspected problems one at a time.
- Think in terms of card replacement only.
- Keep track of *any* unrecorded failure symptoms or unexpected behaviors for future revisions of this guide.
- To test a system, start with a simple hardware configuration and add one card at a time until a failed interface appears or is isolated. Use a simple software configuration and test connectivity using a **ping** test.

Use Table 1-2 as the next step in evaluating hardware. The problems listed are *not* all of the possible failures for each product, but do represent commonly encountered symptoms. Where applicable, possible error messages associated with failure symptoms are also listed.

If you determine that a part or card replacement is required, contact your sales or technical support representative. Specific instructions concerning part or card installation are included with the configuration note provided with the replacement.

If a part replacement appears to solve a problem, make certain to reinstall the suspect part to verify the failure. Remember, if something seems too good to be true, it probably is; *always* double-check a repair.

Table 1-2 Failure Symptoms by Card or Product Type.

Card Type or Part	Failure Symptoms This Card May Cause
CSC-ENVM	<ul style="list-style-type: none"> ■ System is down after running a short time; DC voltages off; blower on. ■ System will not power up; DC voltages off; blower on. ■ Configuration cannot be written to memory; loses memory over time ■ ENVM fails to shut system down even with excessive heat or DC voltage. <p><i>Error Messages</i>—Bad checksum for configuration memory, configuration memory not set up, nonvolatile memory not present.</p>
CSC/4, CSC/3, and CSC/2	<ul style="list-style-type: none"> ■ System will not boot (any combination of LEDs lighted other than green LED lighted only). ■ Multibus cards are not seen. ■ The ciscoBus controller is not seen (CSC/4 and CSC/3). ■ Partial boot only. ■ Random reboot occurs after initial boot. ■ System will autoboot, but cannot boot manually. ■ System will reboot when configuration memory is written. ■ No response from keyboard. <p><i>Error Messages</i>—Parity error, software versus hardware error, local timeout, bus error, wrong interface, emulation line error, software-forced crashes, checksum mismatch error.</p>
CSC-CCTL and CSC-CCTL 2	<ul style="list-style-type: none"> ■ Some or all ciscoBus cards are not seen. ■ No LEDs light. ■ All LEDs light. ■ Wrong number of LEDs light—too many or too few. ■ Some or all Multibus cards are not seen. <p><i>Error Messages</i>—MEMD failure, MEMA failure, ciscoBus daughter controller failure.</p>
CSC-FCI, CSC-C2FCI, CSC-C2FCI and CSC-C2FCIT	<ul style="list-style-type: none"> ■ Not recognized by ciscoBus controller. ■ FDDI ring will not come up. ■ FDDI ring up, but no ping on FDDI ring; intermittent ping; only certain packet sizes will ping. ■ No keyboard response after FDDI ring comes up; lock-up. <p><i>Error Messages</i>—Unknown data error, MEMD/MEMA failure, ciscoBus daughter controller failure.</p>
FDDI Applique (APP-LMM, APP-LMS, APP-LSM, and APP-LSS)	<ul style="list-style-type: none"> ■ FDDI ring will not come up. ■ LEDs are in wrong sequence. ■ FDDI ring will come up in “wrap-mode” only—wrap A or wrap B. ■ No ping through FDDI ring or to address of Unit Under Test (UUT); intermittent ping. ■ FDDI ring will intermittently or constantly transition.

Card Type or Part	Failure Symptoms This Card May Cause
CSC-MEC	<ul style="list-style-type: none"> ■ Card is not seen by ciscoBus controller. ■ Unable to ping on any or some ports; intermittent ping; only certain packet sizes will ping. ■ All LEDs light. ■ No LEDs light. ■ Wrong number of LEDs light. <p><i>Error Messages</i>—Multibus timeout, ciscoBus daughter controller failed, output hung.</p>
MCI and SCI	<ul style="list-style-type: none"> ■ Card is not seen by processor card. ■ No LEDs light. ■ All LEDs light. ■ No ping on any or some ports; DTE will ping and DCE will not ping (or vice versa); intermittent ping; only certain packet sizes will ping. ■ Ports will not initialize—some or all. ■ Will not netboot or ping to network; no ping to address of UUT. ■ MCI-3 cannot see nonvolatile memory (NVRAM) port . <p><i>Error Messages</i>—Local timeout, MEMD failure, MEMA failure, output hang error, bus/ALU failure, configuration memory not set up, excessive input serial error, or Multibus timeouts.</p>
ciscoBus backplane and Multibus backplane	<ul style="list-style-type: none"> ■ Cannot write configuration to memory; no memory access, memory access causes reboot. ■ The ciscoBus cards are not seen. ■ System will not boot or will reboot. ■ No DC voltages—some or all. ■ Bad power supply (caused by shorted backplane).
CSC-R, CSC-R16M, CSC-1R, CSC-2R, and CSC-CTR	<ul style="list-style-type: none"> ■ Card is not seen by processor. ■ No ping to outside address or address of UUT; intermittent ping. ■ No hardware address seen. <p><i>Error Messages</i>—Output hang, beaconing, local timeout, Open failed: lobe test, Multibus timeout.</p>
CSC-M, CSC-MT, CSC-MC, and CSC-MC+	<ul style="list-style-type: none"> ■ NVRAM not seen by MCI-3 (CSC-MC). ■ Configuration cannot be written to memory. ■ Loses memory over time. ■ Configuration and/or Multibus memory wrong size (CSC-MT). <p><i>Error Messages</i>—Bad checksum for configuration memory, configuration memory not set up, nonvolatile memory not present.</p>
Serial appliques	<ul style="list-style-type: none"> ■ Interface up but ping does not work, or intermittent ping functionality. ■ DTE will ping, DCE will not ping (or vice versa). ■ System reboots (with new V.35, suggests bad ground contact). ■ 5V or 12V power supply LEDs indicate no power detected.

Card Type or Part	Failure Symptoms This Card May Cause
IGS and 3000	<ul style="list-style-type: none"> ■ System will not boot. ■ Breaker trips or fuse blows. ■ Constant or partial reboot.
500-CS	<ul style="list-style-type: none"> ■ System will not boot.
4000	<ul style="list-style-type: none"> ■ System will not boot.

Using CiscoWorks to Troubleshoot Your Internet

CiscoWorks is a router management tool that allows you to manage your internet from a central location. You can use CiscoWorks to monitor and troubleshoot complex internetworks. Because CiscoWorks uses the Simple Network Management Protocol (SNMP), it can monitor and control any SNMP device on an internet. CiscoWorks consists of five areas of operation: configuration management (which includes device management), fault management, accounting management, performance management, and security management.

In addition to the basic SNMP management functions, CiscoWorks provides a fully integrated relational database and uses built-in Sun Network Manager (SNM) capabilities to produce a dynamic, user-configurable, visual network map. The automatic map generation features associated with the CiscoWorks Path Tool capabilities can help you visually trace the routes to problem nodes.

Tools that can help you isolate connectivity and performance problems are outlined briefly in the following discussions. Refer to the *CiscoWorks User Guide* for complete details about using CiscoWorks to monitor and control your internetwork.

Using CiscoWorks to Troubleshoot Connectivity Problems

Use the following CiscoWorks fault management applications when troubleshooting connectivity problems in your internet.

- **Device Monitor**—Monitors specific devices for environmental and interface information. Sends event information to SNM that causes a glyph to change state.
- **Path Tool**—Graphically displays a route of the path from a source device to a destination device.
- **Env. Monitor**—Graphically displays the temperature and voltage data from an AGS+ router.
- **Real-Time Graphs**—Monitors the behavior of device interfaces or other network elements suspected of operating in a degraded mode and displays them in a graph.
- **Show Commands**—Enables you to view data similar to output from router EXEC **show** commands.
- **Health Monitor**—Provides information about the health of a device with access to several CiscoWorks applications on one window (including Show Commands and Real-Time Graphs) to monitor router activity.
- **Contacts**—Provides quick access to find your emergency contact person for a particular device.
- **Log Manager**—Enables you to store, query, and delete messages gathered from CiscoWorks applications and Cisco Systems devices on the internetwork.

Using CiscoWorks to Troubleshoot Performance Problems

Use the following CiscoWorks performance management applications when troubleshooting performance problems in your internetwork:

- Device Polling—Probes and extracts data about the condition of your network devices.
- Polling Summary—Views polling data, stops and starts polling.
- Real-Time Graphs—Monitors the behavior of device interfaces or other network elements suspected of operating in a degraded mode and displays them in a graph.
- Path Tool—Graphically displays a route of the path from a source device to a destination device.
- Show Commands—Provides data similar to router EXEC **show** commands output.
- Sybase DWB—Allows you to access the Sybase Data Workbench application to write reports.

Using Third-Party Troubleshooting Tools

This publication emphasizes diagnostic tools provided with the router. However, other troubleshooting tools are also discussed in the symptom modules and scenarios.

In some cases, third-party diagnostic tools can be more useful than integrated tools. For example, enabling a **debug** command can be disastrous in any environment experiencing excessively high traffic levels. Attaching a network or serial analyzer to the suspect network is less intrusive and more likely to yield applicable information without exacerbating load problems for a router.

The following list summarizes some typical third-party troubleshooting tools:

- *Time Domain Reflectometer (TDR)*—A TDR transmits a short pulse of known amplitude and duration down a cable and measures the corresponding amplitude and time delay associated with resultant signal reflections. TDRs are available for all LAN types. Optical TDRs provide a similar test capability for fiber cable.
- *Optical Power Source and Meter*—This device employs an optical power source connected to one end of a fiber cable and a meter placed at the other end to measure optical power. Also called a “light meter,” this device is a cost-effective alternative to an optical TDR.
- *Oscilloscope*—Scopes graphically display signal voltage per unit of time; commonly used to measure voltages on EIA-232 and EIA-422 interfaces.
- *Breakout Box (BOB)*—A BOB displays and monitors status of EIA-232-D interface leads between Data Terminal Equipment (DTE) and Data Circuit-Terminating Equipment (DCE). BOBs are useful in reconfiguring interfaces.
- *LAN Analyzer*—LAN analyzers capture, record, and analyze frames transmitted on a LAN. Analyzers attach to a network just as any node does. These devices also are referred to as protocol analyzers and network analyzers. All analyzers support a range of physical interface specifications (including Ethernet, Token Ring, and FDDI), as well as a spectrum of network protocols (including TCP/IP, Novell IPX, IBM SNA, AppleTalk, DECnet, and ISO CLNS).
- *WAN/Serial Line Analyzer*—WAN protocol analyzers generally focus on WAN/serial line analysis, but can include LAN analysis capabilities. WAN analyzers support a range of physical interfaces (such as EIA-232, EIA-422, EIA-449, T1/E1, CCITT V.35, and CCITT X.21) and protocols (including HDLC, SDLC, Frame Relay, and ISDN).

Troubleshooting Media Problems

Table 1-3 through Table 1-6 summarize general problem-solving guidelines for common media (Ethernet, serial/WAN, Token Ring and FDDI).

Table 1-3 Suggested Actions for Ethernet Problems

Media Problem	Suggested Actions
Errors or noise on Ethernet	Step 1: Use a Time Domain Reflectometer (TDR) to find any unterminated Ethernet cables.
	Step 2: Check host cables and transceiver cables to determine whether any are incorrectly terminated, overly long, or damaged.
	Step 3: Look for a jabbering transceiver attached to a host (may require host-by-host inspection).

Table 1-4 Suggested Actions for Serial Line Problems

Media Problem	Suggested Action
Nonfunctional serial link	Step 1: Use show interfaces serial number command to determine status of interface.
	Step 2: If show interfaces serial number indicates interface up/line protocol up, use the ping command between routers to test connectivity.
	Step 3: If routers do not respond to ping test, follow troubleshooting techniques as discussed in Chapter 7, "Troubleshooting WAN Connectivity."

Table 1-5 Suggested Actions for Token Ring Problems

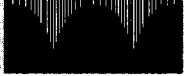
Media Problem	Suggested Action
Nonfunctional Token Ring	Step 1: Use show interfaces token number command to determine status of interface.
	Step 2: If status line indicates that the interface and line protocol are not up, check cable from router to MAU. Make sure that the cable is good; replace if necessary.
	Step 3: If show interfaces token number indicates interface up/line protocol up, use the ping command between routers to test connectivity.
	Step 4: If the remote router does not respond, check the ring specification on all nodes attached to the Token Ring backbone. Ring speed for all must be the same.

Media Problem	Suggested Action
	<p>Step 5: If necessary, modify ring speed specifications for clients, servers, and routers.</p> <p>Step 6: Use the ring speed command to modify ring speed configuration for IGS/TR. Change jumpers as needed for modular router platforms. Refer to your system's hardware installation and maintenance manual for more information about ring speed specification.</p>

Table 1-6 Suggested Actions for FDDI Problems

Media Problem	Suggested Actions
Nonfunctional FDDI ring	<p>Step 1: Use the show interfaces fddi number command to determine status of interface.</p> <p>Step 2: If show interfaces fddi number indicates interface up/line protocol up, use the ping command between routers to test connectivity.</p> <p>Step 3: If interface is up and line protocol is up, make sure the MAC addresses of upstream and downstream neighbors is as expected. If all zeros appear in either of the address fields for these neighbors, a physical connection problem is likely.</p> <p>Step 4: In this case, (or if status line does <i>not</i> indicate interface up/line protocol up), check connections at patch panel or connectivity between using an optical TDR or light meter. Ensure that signal strength is within specification.</p>

CLICCO SYSTEMS



1

Chapter 2

Connectivity Problem Scenarios

2

Connectivity Scenario Overview 2-2

Apple Service Availability Scenario 2-3

- Symptoms 2-3
- Environment Description 2-4
- Diagnosing and Isolating Problem Causes 2-5
- Problem Resolution Process 2-6
 - Looking for a ZIP Storm 2-7
 - Isolating Duplicate Network Addresses 2-7
 - Identifying a Phase 1/Phase 2 Rule Violation 2-8
 - Establishing Printer Service over Internet 2-10
- Problem Solution Summary 2-12

Concurrent Routing and Source-Route Bridging Connectivity Problems 2-15

- Symptoms 2-15
- Environment Description 2-15
- Diagnosing and Isolating Problem Causes 2-16
 - Finding Missing Multiring Subcommands 2-16
 - Looking for a Misconfigured IP Address 2-17
 - Checking the End Systems 2-17
- Problem Solution Summary 2-19

Translational Bridging, SRT, STUN, and SDLLC Connectivity Problems 2-21

- Symptoms 2-21
- Environment Description 2-21
- Diagnosing and Isolating Problem Causes 2-23
 - Detecting Incompatibilities Between End Systems and Intermediate Systems 2-23
 - Detecting SRT/SRB Incompatibilities 2-24
 - Resolving Vendor Code Mismatch Problems 2-26
 - Finding Missing Multiring Subcommands 2-26
 - Enabling Access to the AS/400 on Ring #2 2-27
- Problem Solution Summary 2-27

Novell Network Server Connectivity Scenario 2-33

- Symptoms 2-33
- Environment Description 2-34
- Diagnosing and Isolating Problem Causes 2-34

- Checking Physical Attachment of Clients to Network 2-35
- Checking Physical Attachment of Servers to Network 2-36
- Enabling Novell IPX Routing 2-36
- Determining Whether SAP Updates Are Disabled 2-36
- Checking Novell Network Number Specifications 2-37
- Checking Router Interface Status 2-38
- Checking for Limited-User Version of NetWare 2-38
- Checking for Encapsulation Mismatch 2-38
- Checking for Access List Problems 2-39
- Checking for Nonunique MAC Addresses on Routers 2-39
- Checking for Misconfigured Helper Addresses 2-40
- Finding a Backdoor Bridge 2-41

Problem Solution Summary 2-42

TCP/IP Route Redistribution and Access Control Scenario 2-45

Symptoms 2-45

Environment Description 2-46

Diagnosing and Isolating Problem Causes 2-46

- Isolating Router Software Configuration Problems 2-46

Problem Solution Summary 2-49

X.25 WAN Router Initial Installation Problems 2-51

Symptoms 2-51

Environment Description 2-51

Diagnosing and Isolating Problem Causes 2-52

- Isolating Serial Hardware and Media Problems 2-53

- Isolating Interface, LAN, and Local Host Configuration Problems 2-56

- Isolating Router Software Configuration Problems 2-57

Problem Solution Summary 2-61

Chapter 2

Connectivity Problem Scenarios

2

This chapter presents problem-solving *scenarios* focusing on identifying, isolating, and solving problems that block connectivity in internetworks. These kinds of problems also are referred to as *reachability* problems.

The problem-solving scenarios addressed here provide details concerning specific situations and illustrate the process of problem isolation and resolution. The scenarios provided here span different protocols, media, and problem types. The objective of the scenarios is to illustrate a problem-solving method based on the general problem-solving model defined in Chapter 1, “Troubleshooting Overview.” These scenarios are *composites* of real-world situations.

Each scenario includes the following components:

- Symptom statement
- Internetworking environment description
- Problem isolation discussion and process
- Solution summary

Subsequent chapters present a series of *symptom modules* that provide snapshots of common symptoms, possible causes, and suggested actions.

More details concerning scenarios and symptom modules are provided in the section “How to Use This Publication,” in Chapter 1.



Caution: Throughout this and other chapters, the use of **debug** commands is suggested for obtaining information about network traffic and router status. Use these commands with great care. In general, it is recommended that these commands only be used under the direction of your router technical support representative when troubleshooting specific problems. Enabling debugging can disrupt operation of the router when internets are experiencing high load conditions. When you finish using a **debug** command, remember to disable it with its specific **undebug** command or with the **undebug all** command.

Connectivity Scenario Overview

Connectivity problems manifest themselves in many ways. Some examples are users' inability to make terminal connections, known routes not appearing in routing tables, or file servers failing to respond to boot requests. Similarly, certain *symptoms* (such as users being unable to connect to hosts) may result from a variety of problems.

This chapter presents a series of situational discussions and includes application of various diagnostic tools. Every possible scenario obviously cannot be covered. Indeed, the scenarios included here only scratch the surface of possible situations. However, certain common themes typically tie all connectivity problems together. This chapter illustrates the use of troubleshooting tools and techniques to identify those common themes.

The following problem-solving scenarios are presented in this chapter:

- **AppleTalk Service Scenario**—This scenario presents several common problems that can block access to servers and services on an AppleTalk internet.
- **Concurrent Routing/SRB Scenario**—This scenario features both routed and bridged protocols, and illustrates workstations that are unable to access target hosts and resources.
- **IBM SNA Scenario**—This scenario outlines potential implementation problems associated with translational bridging, SRT, SDLC transport, and SDLC-to-LLC translation.
- **Novell Network Server Connectivity Scenario**—This scenario illustrates some of the more common problems that can impair Novell server access over an IPX-based routed internet.
- **TCP/IP Route Redistribution and Access Control Scenario**—This scenario focuses on the issue of balancing connectivity and security.
- **X.25 WAN Router Initial Installation Scenario**—This scenario illustrates blocked connectivity over a newly installed router connected to an X.25 private WAN.

Apple Service Availability Scenario

In recent years, AppleTalk-based internetworks have grown in size and scope of implementation. Once viewed as a toy protocol, AppleTalk has been stretched to allow handling of more nodes and sharing of services in larger internets. Along with these larger-scale and more complex implementations have come some of the implementation headaches common to any multivendor, enterprise internet. This scenario focuses on several common problems that can block access to servers and services on an AppleTalk internet.

Symptoms

In this case, a number of local networks are segmented with routers, while a remote network is linked over a serial line (refer to Figure 2-1).

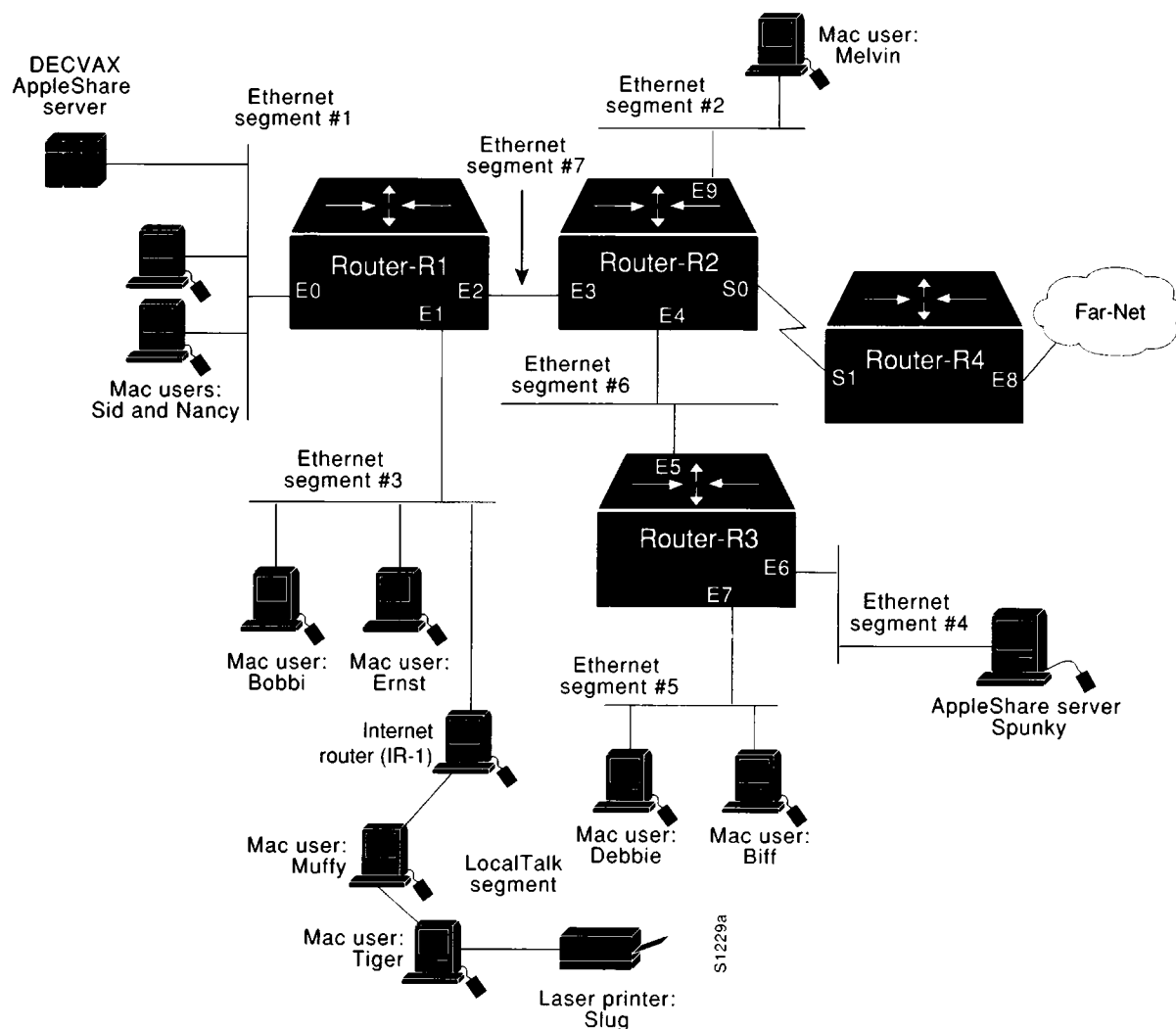


Figure 2-1 Initial AppleTalk Connectivity Scenario Map

Assume that the following symptoms were reported for this AppleTalk internet:

1. Mac User Melvin on Ethernet Segment #2 reports that the laser printer Slug (attached to the LocalTalk network connected to IR-1) is not visible on his Chooser.
2. DEC VAX-based AppleShare Server on Ethernet Segment #1 is not visible to any users except Mac users Debbie and Biff on Ethernet Segment #5.
3. AppleShare server Spunky on Ethernet Segment #4 is sometimes visible in the Choosers of Mac users in this internet, but no one can access services on that server. Although users on the same network as Spunky can see local services, they find that it is difficult to access offnet services.

There are a number of problems that might lead to these symptoms. The first step is to characterize this internet's configuration and then develop a list of likely suspect problems.

Environment Description

Some relevant facts regarding the internetworking environment shown in Figure 2-1 can be summarized as follows:

- Three Cisco routers (Router-R1, Router-R2, and Router-R3) and a non-Cisco internet router (IR-1) provide interconnection between local Ethernet segments and a LocalTalk network attached to IR-1.
- Remote service is provided via Router-R2 and the remotely located (Cisco) Router-R4 to an AppleTalk network (Far-Net) that is not controlled by local network administration.
- Mac users in the same zone as the DEC VAX can see all zones and can access offnet services.
- Users on all the local networks can access AppleTalk services on directly connected network cables.
- The routers in this internet are in the process of being converted from Phase 1 support to Phase 2 support.
- The only other protocol used in this internet is TCP/IP.
- With the exception of one LocalTalk segment, local networks are IEEE 802.3 Thin Ethernets; the serial link is a dedicated T1 link (1.544 Mbps).
- The network applications intended to run over the T1 line include typical Apple network services.

Diagnosing and Isolating Problem Causes

Given the situation, a number of problems could explain reported symptoms.

The following problems are likely candidates for symptom #1 (laser printer Slug on Ethernet Segment #3 is reported as not visible on Chooser by Mac User Melvin on Ethernet Segment #2.):

- Misconfigured router (Router-R1 or IR-1).
- Ethernet port on Router-R1 is shut down.

The following problems are likely candidates for symptom #2 (DECVAX-based AppleShare Server on Ethernet Segment #1 is not visible to any users except users on Ethernet Segment #5—a nonextended network):

- Duplicate network number
- Phase 1/Phase 2 internetworking rule violation
- Network or port configuration mismatch

The following problems are likely candidates for symptom #3 (AppleShare server Spunky on Ethernet Segment #4 is sometimes visible in Chooser of Mac users in this internet, but no one can access services on that server):

- Duplicate network number
- Zone Information Protocol (ZIP) storm

Once a possible problem list is determined, each potential cause must be systematically analyzed. The following discussion considers the possible problems listed and illustrates resolution of discovered problems.

Before continuing with this process, it will be useful to map out the assignment of network numbers/cable ranges and zones (or zone lists) associated with the internet. Figure 2-2 illustrates the known network numbers, cable ranges, and zones associated with this internet.

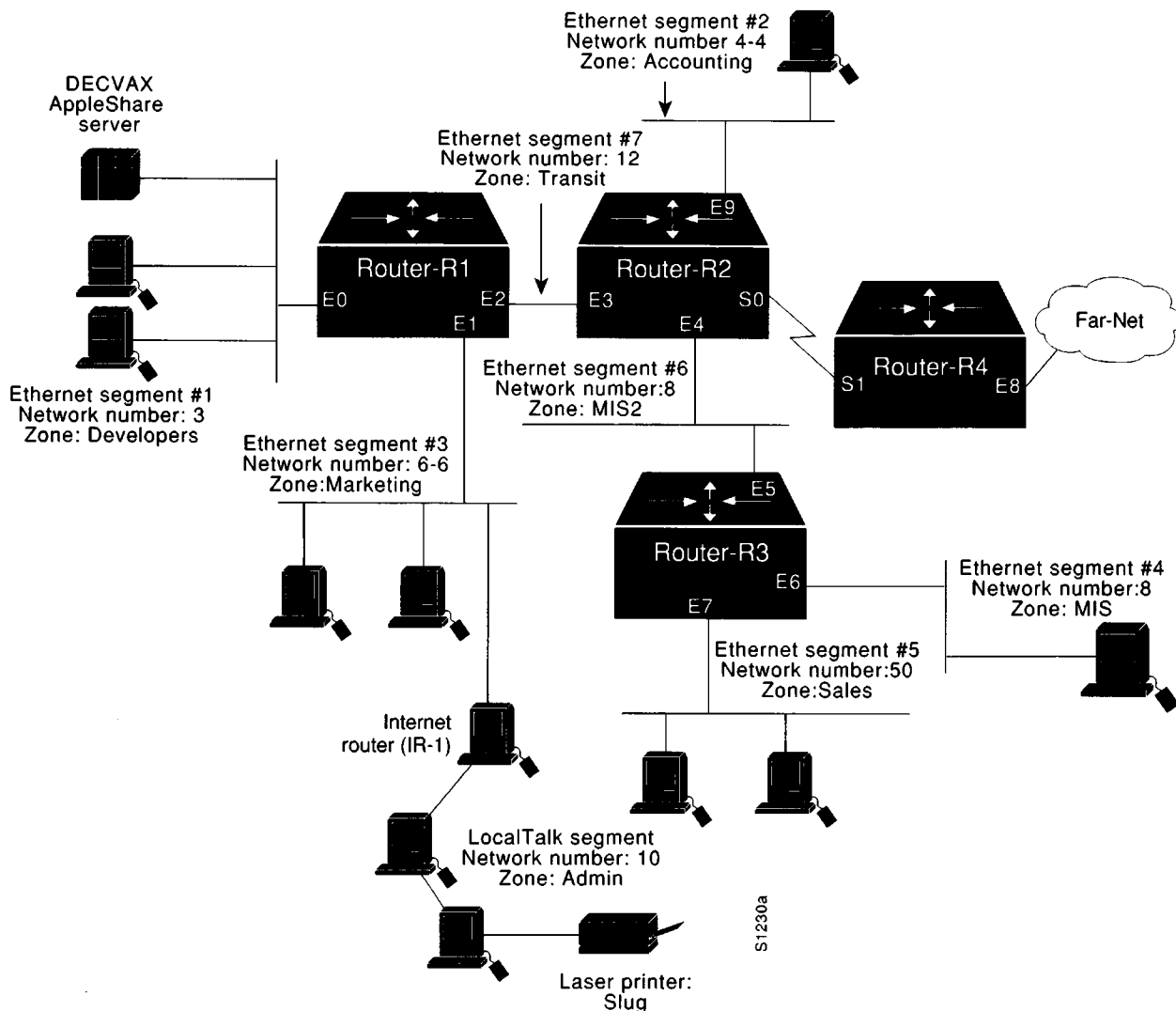


Figure 2-2 AppleTalk Zone and Network Number/Cable Range Assignments

Problem Resolution Process

This analysis starts by considering the problem list associated with Spunky's intermittent availability (symptom #3). At the same time, since the DEC VAX problem shares a possible cause with the Spunky access problem, the analysis evaluates the possibility of a common problem causing both symptoms. After that, the analysis steps through the list of possible problems until all possible causes are exhausted.

Looking for a ZIP Storm

It is not unusual to start with a possible problem because it is *easy* to diagnose. With this in mind, first consider the possibility of a ZIP storm.

Step 1: To detect a ZIP storm, first examine network activity with the **show appletalk traffic** command.

Look for “ZIP Requests” in the resulting display. Repeat this command after about 30 seconds or so. If the number is greater than 10 and increasing, there is likely to be a ZIP storm.

Step 2: If you observe an apparent ZIP storm, use the **show appletalk route** command and look for a network that shows up in the table but has “no zone set” indicated for its zone listing. If such a listing appears, determine why the node is not responding to ZIP requests.

For this case, assume that no unusual number of ZIP requests appear and you eliminate a ZIP storm as a cause for symptom #3. All symptoms are still being experienced.

Isolating Duplicate Network Addresses

The next possible cause for both symptom #2 and symptom #3 is the existence of duplicate network numbers in the internet. Unfortunately, these are not usually easy to find.

Step 1: First, use **show appletalk interface ethernet 6** on Router-R3 to obtain the AppleTalk network number for the local network. In this case, the (non-extended) network number is 2. Figure 2-3 illustrates a typical output for this command.

```
Ethernet 6 is up, line protocol is up
  AppleTalk address is 2.12
  AppleTalk zone is "Far-Away"
```

Figure 2-3 Show AppleTalk Interface Ethernet 6 Command Output

Step 2: Next, disable AppleTalk using the **no appletalk routing** global configuration command as illustrated in Figure 2-4.

```
Router-R3# configure terminal
no appletalk routing
<Ctrl-Z>
```

Figure 2-4 Disabling AppleTalk for the Router

If there are no duplicate network numbers (another network number 2), **no appletalk routing** results in network number 2 being aged out of all routing tables in the internet.

Step 3: To determine whether this happens, perform successive **show appletalk route 2** commands on Router-R3 until the hop count stabilizes (indicating that a duplicate does exist), or until the route ages out (indicating that a duplicate does not exist).

If there is a duplicate, network 2 will not age out, but instead, appears as a learned route from some other interface. Figure 2-5 illustrates how this change is registered in the **show appletalk route 2** display.

**Indicates network 2 is now
learned via Ethernet5**

Codes: R - RTMP derived, C - connected, 95 routes in internet

```
R Net 2 [2/G] via 8.2, 3 sec, Ethernet5, zone Far-Away
Route installed 79:43:39
Current gateway: 8.2, 2 hops away, updated 3 secs ago
Zone list provided by 8.2
Route has been updated since last RTMP was sent
Valid zones: "Far-Away"
```

Figure 2-5 Example Show AppleTalk Route 2 Command Output

Figure 2-5 indicates the neighbor from which the location of the duplicate was learned. Since IP also is enabled in this internet, you can pinpoint the duplicate network number by connecting to the indicated neighbor. Telnet to the indicated neighbor (here at *network.node* address 8.2), and remember to use the IP address or the router's IP host name (in this case assume Router-R2).

Step 4: Once a connection is made to the neighbor, repeat the **show appletalk route 2** command and examine the resulting output for the location of network number 2. Iterate this process until the display indicates that the network is *directly* connected.

Step 5: When the network is shown as directly connected, you have found the duplicate network number location. Now, you must change one of the routers (Router-R3 or the found router), as well as any other routers connected to the suspect network.

Assume that doing this solves Symptom 3; offnet Macintosh users in the internet now can access AppleShare server Spunky (after Ethernet interface E6 on Router-R3 has been restored to service).

However, users still cannot access the DEC VAX AppleShare server, and the laser printer Slug remains inaccessible.

Identifying a Phase 1/Phase 2 Rule Violation

It is quite possible that there is yet another duplicate network number in the internet, resulting in the DEC VAX being unavailable as an AppleShare server.

However, remember that VAX AppleShare service is accessible to Mac users Biff and Debbie on Ethernet Segment #5 (network number 50). This suggests that it is unlikely that access to the VAX is limited by the same problem resolved in the prior step. It also rules out port configuration mismatch as a problem, as well as duplicate network numbers, because Router-R1 and Router-R3 agree about network configuration (network number/cable range and zone/zone list). This leaves a Phase 1/Phase 2 rule violation as the remaining identified possible cause.

Step 1: To determine whether this is the problem, use the **show appletalk global** command. Figure 2-6 illustrates the output of this command when the network is in compatibility mode. However, this display indicates that the internet is *not* in compatibility mode when a Phase 1/Phase 2 rule violation exists. A rule violation is said to exist when any node has a configuration that does not conform to the following rules:

- There can be no wide cable range specifications in the Phase 2 extended portion of the internet (cable ranges must be specified to include only a single network number, such as 2-2 or 10-10).
- Multiple zones cannot be assigned to networks or cable ranges.

AppleTalk global information:

Internet is compatible with older, AT Phase1, routers.

There are 95 routes in the internet.

There are 30 zones defined.

Logging of significant AppleTalk events is disabled.

ZIP resends queries every 10 seconds.

RTMP updates are sent every 10 seconds.

RTMP entries are considered BAD after 20 seconds.

RTMP entries are discarded after 60 seconds.

AARP probe retransmit count: 10, interval: 200.

AARP request retransmit count: 5, interval: 1000.

DDP datagrams will be checksummed.

RTMP datagrams will be strictly checked.

RTMP routes may not be propagated without zones.

IPTalk uses the udp base port of 768 (Default).

Alternate node address format will not be displayed.

Access control of any networks of a zone hides the zone.

Names of local servers will be queried every 60 seconds.

Lookups will be generated for server types:

IPADDRESS, IPGATEWAY

This field indicates when violations exist. In this case, it indicates that the internet is in compliance with compatibility rules.

Figure 2-6 Standard Output of Show AppleTalk Global Command

Step 2: Next, use the **show appletalk neighbor** command at Router-R1 to identify the specific neighboring router that requires compatibility mode. Figure 2-7 illustrates such a listing.

**Indicates that the neighbor
requires compatibility mode
and does not support extended
networks.**

```

AppleTalk neighbors:
  3.3      Ethernet0, uptime 57:47:23, 0 secs
           Neighbor requires compatibility mode
  4160.2   Ethernet1, uptime 90:20:11, 0 secs
           Neighbor has restarted 3 times in 40:12:34.
           Neighbor update is overdue.
  4160.4   Ethernet1, uptime 120:53:54, 435137 secs
           Neighbor has restarted 2 times in 121:01:42.
           Neighbor update is overdue.
  4160.41  Ethernet1, uptime 195:28:14, 701994 secs
  
```

Figure 2-7 Example Show AppleTalk Neighbor Display Output

Step 3: In this case, the neighbor needing compatibility mode is the VAX itself. At this point, there are two solution options: upgrade the VAX AppleShare server or use the **appletalk proxy-nbp** global configuration command.

This command creates what is in effect a virtual network off Router-R1. The command to implement would be as follows:

appletalk proxy-nbp 200 Developers

Note that no router can have the same network number defined as a proxy network, and the specified network number cannot be associated with a physical network.

Adding **appletalk proxy-nbp** forces Router-R1 to send the proper NBP Lookup Packet out all networks for the zone named “Developers.” Using this command then resolves that problem of access to the VAX AppleShare server from extended networks.

However, laser printer Slug is still not accessible from Mac user Melvin on Ethernet Segment #2.

Establishing Printer Service over Internet

Two possible causes were cited for blocking availability to Slug: either the Router-R1 port is down, or Router-R1 or IR-1 has a configuration problem. Assume that Bobbi and Ernst (on extended network 6-6, zone Marketing) can now access offnet zones and service over Router-R1, but cannot see services on the other side of IR-1. This suggests Router-R1 is probably operational and that the problem probably is with IR-1.

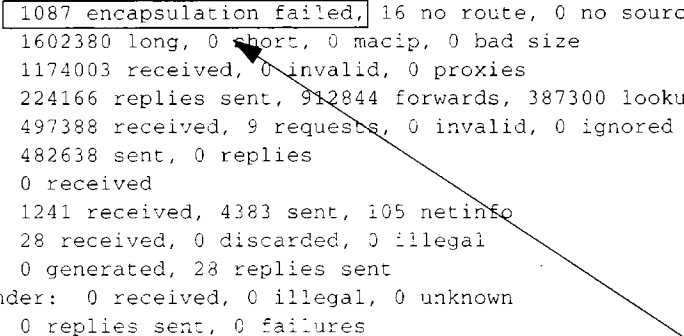
Step 1: Use the **show appletalk neighbor** command to determine whether Router-R1 can see IR-1. Look for any neighbors. If IR-1 has a configuration problem, it probably will not appear in the neighbor listing.

Step 2: Before proceeding with any further configuration analysis, verify that the cabling at IR-1 is intact. Try **show appletalk neighbor** from Router-R1 again. If router IR-1 still does not appear in the neighbor listing at this point, it is safe to

suspect that IR-1 is a Phase 1 router and will require upgrading to support of AppleTalk Phase 2 to operate in this internet.

Step 3: For further evidence, use the **show appletalk traffic** command and look for encapsulation failures. If you see more than 100 encapsulation failures, it might suggest Phase 1/Phase 2 problems, and again bolsters the hypothesis that IR-1 is the problem in this case. Figure 2-8 illustrates the output of the **show appletalk traffic** command.

```
AppleTalk statistics:
  Rcvd: 1807514 total, 0 checksum errors, 7541 bad hop count
        1596186 local destination, 0 access denied
        0 for MacIP, 0 bad MacIP, 0 no client
        0 port disabled, 0 no listener
        0 ignored, 0 martians
  Bcast: 808385 received, 560408 sent
  Sent: 1530871 generated, 7422 forwarded, 222001 fast forwarded 24408 loopback
        0 forwarded from MacIP, 0 MacIP failures
        1087 encapsulation failed, 16 no route, 0 no source
  DDP: 1602380 long, 0 short, 0 macip, 0 bad size
  NBP: 1174003 received, 0 invalid, 0 proxies
        224166 replies sent, 912844 forwards, 387300 lookups, 1223 failures
  RTMP: 497388 received, 9 requests, 0 invalid, 0 ignored
        482638 sent, 0 replies
  ATP: 0 received
  ZIP: 1241 received, 4383 sent, 105 netinfo
  Echo: 28 received, 0 discarded, 0 illegal
        0 generated, 28 replies sent
  Responder: 0 received, 0 illegal, 0 unknown
        0 replies sent, 0 failures
  AARP: 384 requests, 595 replies, 1341 probes
        30 martians, 0 bad encapsulation, 0 unknown
        772 sent, 0 failures, 276 delays, 1087 drops
  Lost: 0 no buffers
  Unknown: 0 packets
  Discarded: 826 wrong encapsulation, 0 bad SNAP discriminator
```



Field reporting encapsulation failures

Figure 2-8 Example Show AppleTalk Traffic Display Output

Step 4: To verify that IR-1 is a Phase 1 router, first bring up Router-R1 in *discovery mode*. This is done by (temporarily) setting the AppleTalk address for interface Ethernet1 to 0.0 using the **appletalk address** interface subcommand. Once this configuration is done, Router-R1 attempts to acquire configuration information for that cable from an operational Phase 1 router.

Making this change has the following effects:

- Router-R1 interface Ethernet1 comes up as a nonextended network.
- All nodes on the attached network cable range 6-6 are isolated.

However, this confirms that IR-1 is Phase 1 router (confirmation also can be done using the IR-1 configuration utility).

Step 5: To resolve this access problem, IR-1 must be upgraded to be a Phase 2 AppleTalk router, and the Cisco router must be reconfigured back to its original state (Ethernet interface E1 had an extended network cable range of 6-6).

Problem Solution Summary

This scenario focused on diagnosing blocked service access in AppleTalk internets. Modifications discussed in this scenario included the following:

- Upgrading a Phase 1-only router to support Phase 2 removed blocked print service.
- Using the **appletalk proxy-nbp** command allowed access to a DEC VAX-based AppleShare server requiring Phase 1 compatibility.
- Eliminating duplicate network numbers ensured access to AppleShare server Spunky.

Figure 2-9 illustrates an example final configuration listing for Router-R1, obtained using the **write terminal EXEC** command, where **appletalk proxy-nbp** has been added.

```

Current configuration:
version 9.1
!
hostname Router-R1
!
enable-password toYNetgmn
!
appletalk routing
!
interface Ethernet 0
ip address 131.108.29.18 255.255.255.0
ip helper-address 131.108.13.111
ip helper-address 131.108.1.255
ip helper-address 131.108.13.255
keepalive 5
appletalk address 3.24
appletalk zone Developers
!
interface Ethernet 1
ip address 131.108.160.18 255.255.255.0
ip helper-address 131.108.1.255
keepalive 5
appletalk cable-range 6-6 6.19
appletalk zone Marketing
!
interface Ethernet 2
ip address 131.108.161.18 255.255.255.0
ip helper-address 131.108.1.255
keepalive 5
appletalk address 12.90
appletalk zone Transit
!
ip route 131.108.171.0 255.255.255.0 131.108.165.73
ip route 131.108.170.0 255.255.255.0 131.108.165.73
!
!
appletalk name-lookup-interval 60
appletalk lookup-type IPADDRESS
appletalk lookup-type IPGATEWAY
appletalk proxy-nbp 200 Developers
!
line aux 0
login
line vty 0 4
login
line con 0
exec-timeout 0 0
password k1EwdGD
line aux 0
no exec
exec-timeout 0 0
password k1EwdGD
line vty 0
exec-timeout 0 0
password k1EwdGD
!
end

```

Figure 2-9 Complete Router-R1 Final Configuration

Concurrent Routing and Source-Route Bridging Connectivity Problems

With multiprotocol internets, the chances of misconfiguration resulting in connectivity loss are substantially greater than with single-protocol networking environments. Along with the added efficiency and flexibility of multiprotocol internets comes an added level of management complexity.

The following connectivity-related scenario features both Novell and Sun networking systems sharing access to resources over Token Ring and serial media. This scenario illustrates problems facing internets characterized by concurrent bridging and routing.

Symptoms

Consider a corporate network composed of Token Ring segments partitioned with source-route bridges (SRBs) as illustrated in Figure 2-10. Here, the PCs on Ring 4 are unable to connect to Novell servers on Rings 2 and 3, while a PC on Ring 3 cannot communicate with the Sun file server on Ring 4.

Environment Description

Figure 2-10 illustrates a map of the environment discussed in this case. The following summarizes the relevant elements of this internetworking environment:

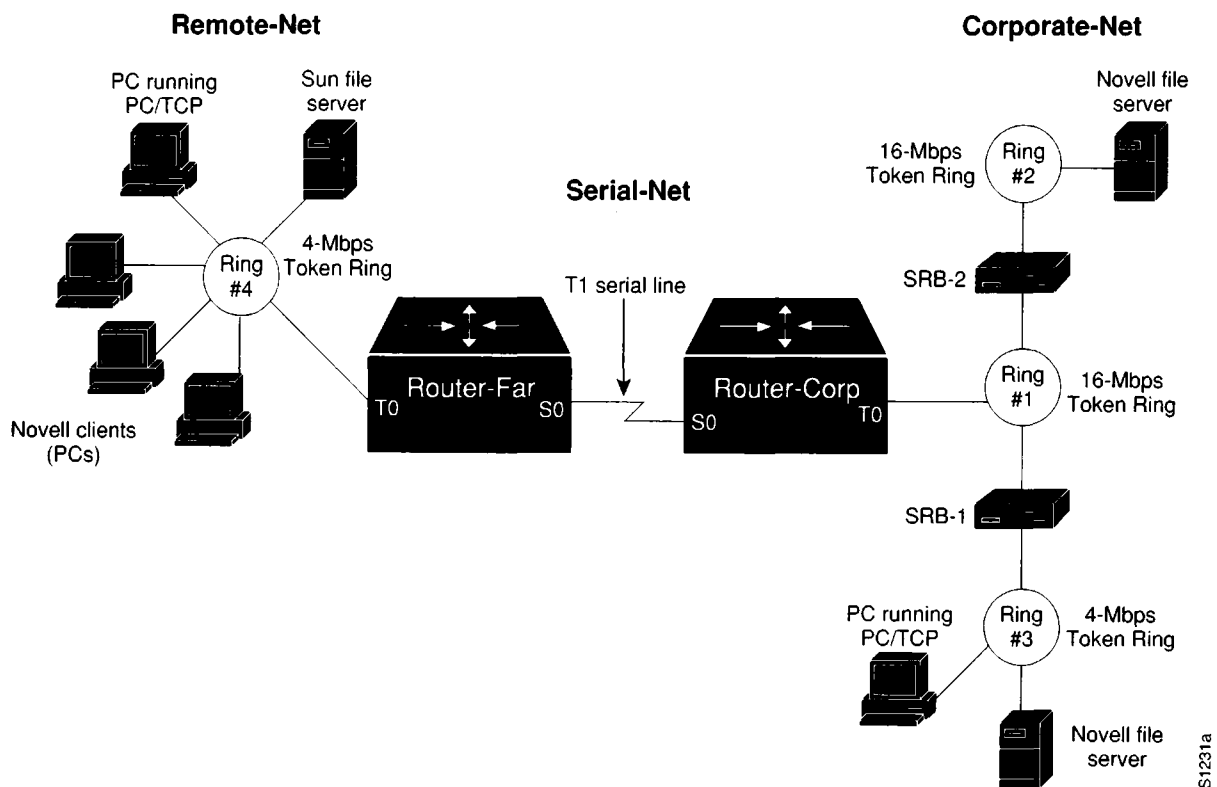


Figure 2-10 Initial SRB/Routing Internet Problem Environment

- The primary corporate network (Corporate-Net) consists of two 16-Mbps and a single 4-Mbps Token Rings separated by non-Cisco SRBs (SRB-1 and SRB-2).
- Users on a 4-Mbps Token Ring at a remote sales office (Remote-Net) are linked to the Corporate-Net over a T1 service (Serial-Net), with routers (Router-Corp and Router-Far) providing routing service for both TCP/IP and Novell IPX traffic between Corporate-Net and Remote-Net.
- The LANs are all IEEE 802.5 Token Rings.
- The network applications running over the WAN include X-Windows, file transfer (FTP), mail (SMTP), Novell NetWare file service, and virtual terminal connections (Telnet).

Diagnosing and Isolating Problem Causes

Given the situation, the following possible problems are the most likely candidates for interconnection failure:

- Missing **multiring** interface subcommand
- Misconfigured IP network addresses
- No source-route bridging driver on a Novell server
- Software bug on some network device

The next step is to eliminate each potential cause as the problem source and then test the network to determine whether it is operational. The following discussion works through the process of problem isolation.

Finding Missing Multiring Subcommands

Given the difficulties being experienced, problems with the router configurations are definite possibilities. In particular, if routed protocols are not making it through an environment consisting of SRBs, look for missing **multiring** interface subcommands. The following steps outline actions to diagnose and remedy potential configuration problems in this kind of environment.

- Step 1:** Using the **write terminal** command on the two routers connected to the T1 serial line, look for a **multiring** interface subcommand for *each* routed protocol, or the **all** keyword option (applied to the Token Ring interfaces).
- Step 2:** Assuming the **multiring** command is not included or does not cover a particular protocol that is being routed (and subsequently bridged over the SRB as in this scenario), be sure to make any required changes. Figure 2-11 illustrates a specification of the **multiring** command that generates RIFs for IP frames but not for Novell IPX frames. Refer to the *Router Products Configuration and Reference* publication for more information about using the **multiring** command.

```
!  
interface tokenring 0  
multiring ip  
ip address 131.108.2.4 255.255.255.0  
novell network 33  
!
```

Figure 2-11 Example Multiring Command Specification

Looking for a Misconfigured IP Address

Another potential configuration-related problem rests with specifying IP network addresses. If incorrectly specified, a discontinuous network space can be created, resulting in a complete stoppage of all IP traffic at the point of discontinuity.

In this scenario, assume Token Rings 1, 2, 3, and 4 are all configured to be on subnet 131.108.1.0. The interfaces attached to the serial line linking the two sites are assigned IP addresses 192.1.100.1 (Router-Far) and 192.1.100.2 (Router-Corp).

The discontinuity in this example results from the separation of segments in the same subnet (the four Token Rings) by a segment that belongs to a different major network (the serial network).

Step 1: Use the **write terminal EXEC** command to determine the address specifications associated with the Token Rings and serial lines to which the routers are attached.

Step 2: There are two options for remedying this situation:

- Reconfigure the IP address assignments for the serial lines so that both interfaces attached to the link belong to the same major network as the Token Rings.
- Assign different network numbers to all three networks (Remote-Net, Serial-Net, and Corporate-Net).

Note: Refer to the *Router Products Configuration and Reference* publication for more information about assigning IP addresses and using subnet addressing.

Checking the End Systems

The end systems (PCs) attached to the various rings represent another likely set of problem sources in this connectivity scenario. The following steps outline actions to diagnose and remedy potential problems associated with the end systems in this kind of environment.

Step 1: Check the end systems for source-route bridge drivers.

Step 2: Reconfigure the end systems or swap out for systems that have the ability to handle SRB.

- Step 3:** In addition to missing SRB drivers, end systems may be unable to participate in protocol exchanges because of software problems (bugs). To isolate this kind of problem in a TCP/IP environment, **ping** the end station.
- Step 4:** If there is no response, use the **show rif** and **show arp** EXEC commands to determine the hardware address of the end station in the ARP and RIF tables. Figure 2-12 and Figure 2-13 illustrate the output of the **show rif** and **show arp** commands, respectively, when an end station is properly listed.

```
Codes: * interface, - static, + remote
Hardware Addr  How   Idle (min)   Routing Information Field
5C02.0001.4322 rg5      -      0630.0053.00B0
5A00.0000.2333 TR0      3      08B0.0101.2201.0FF0
5B01.0000.4444 -        -        -
0000.1403.4800 TR1      0      -
0000.2805.4C00 TR0      *      -
0000.2807.4C00 TR1      *      -
0000.28A8.4800 TR0      0      -
0077.2201.0001 rg5      10     0830.0052.2201.0FF0
```

Figure 2-12 Example Output of Show RIF EXEC Command

Protocol	Address	Age (min)	Hardware Addr	Type	Interface
Internet	131.108.6.65	218	0000.0c02.710b	SNAP	Fddi0
Internet	131.108.6.69	-	0000.0c02.7aae	SNAP	Fddi0
Internet	131.108.134.69	-	0000.0c00.c0d3	ARPA	Ethernet1
Internet	131.108.181.69	-	0000.3040.e028	SNAP	TokenRing2
Internet	131.108.181.69	215	0000.3030.ee2b	SNAP	TokenRing2

Figure 2-13 Example Output of Show ARP EXEC Command

- Step 5:** If the end station does not appear in the table, use the **clear rif** and **clear arp** commands. Set the RIF timeout to a small value and ping the end station at intervals greater than the RIF timeout to see if the end station can respond.
- Step 6:** If the end station does not respond, use a network analyzer, such as a Network General *Sniffer*, to look for the end system's response to the router's XID-to-NULL SAP packet (DSAP value of 00).

The default timeout for ARP table entries is much larger than the RIF entries (such as four hours for ARP and 15 minutes for RIF). The first time that a station is pinged, there are no ARP or RIF table entries for its hardware address, so both entries are updated with the ARP response from the end station. After the default timeout for RIF, the RIF entry is cleared, whereas the ARP entry remains. When this situation arises, if the end station is pinged again, the router generates an XID packet and sends it to the destination hardware address of the end station using a NULL SAP value (DSAP value of 00) to find the RIF.

- Step 7:** If you *do* see the router XID-to-NULL SAP packet, but the end station is unable to respond, there is probably a problem with the end station (host) SRB software and you must upgrade the software on the end system.

(In one case, there was a bug in the IBM RS/6000 where an RS/6000 would not reply to an XID sent using NULL SAP.)

Note: If the PC/workstation is not responding to the XID-to-NULL SAP packet (DSAP value of 00), and you are unable to upgrade software on the end system, try making the ARP timeout on the end system a little less than the RIF timeout. This causes the RIF and ARP to time out at the about the same time and forces the routers to ARP in place of sending XID-to-NULL SAP.

Problem Solution Summary

Topics covered in this scenario addressed a number of common SRB and routing problems encountered in IBM internets. Procedures discussed included the following:

- Adding missing **multiring** interface subcommands to Router-Far and Router-Corp Token Ring interfaces to allow routing of protocols over multiple Token Rings in networks including SRBs.
- Ensuring that the IP addressing of all interfaces created a contiguous network addressing scheme.
- Finding and reconfiguring or replacing Novell end systems that did not include source-route bridging drivers.
- Using integrated router and third-party diagnostic tools to find software bugs on a network device.

Figure 2-14 and Figure 2-15 provide relevant configuration listings for Router-Corp and Router-Far. These configurations illustrate changes required to ensure proper RIF updating and a contiguous network addressing scheme.

```
!Router-Corp configuration:
!
source-bridge ring-group 3
source-bridge remote-peer 3 tcp 150.136.139.1
source-bridge remote-peer 3 tcp 150.136.139.2
!
novell routing 0000.3040.d065
!
interface Serial 0
ip address 150.136.139.1 255.255.255.0
novell network CC
!
interface TokenRing 0
ip address 131.108.1.1 255.255.255.0
ring-speed 16
novell network AA
source-bridge 1 2 3
source-bridge spanning
multiring ip
multiring novell
!
router igrp 109
network 150.136.0.0
network 131.108.0.0
```

Figure 2-14 Relevant Router-Corp Final Configuration

```

!Router-Far configuration:
!
source-bridge ring-group 3
source-bridge remote-peer 3 tcp 150.136.139.1
source-bridge remote-peer 3 tcp 150.136.139.2
!
novell routing 0000.3040.a043
!
!
interface Serial 0
ip address 150.136.139.2 255.255.255.0
novell network CC
!
interface TokenRing 0
ip address 131.108.2.1 255.255.255.0
ring-speed 16
novell network BB
source-bridge 4 5 3
source-bridge spanning
multiring ip
multiring novell
!
!
router igrp 109
network 150.136.0.0
network 131.108.0.0
!

```

Figure 2-15 Relevant Router-Far Final Configuration

Translational Bridging, SRT, STUN, and SDLLC Connectivity Problems

Cisco's IBM connectivity options range from support for source-route bridging (SRB) and source-route transparent (SRT) bridging to translational bridging and SDLC transport over TCP/IP. Thus, network managers can tailor router configurations to the specific needs of existing networks and then reconfigure routers to respond to network changes.

The scenario that follows illustrates some of the pitfalls that you may encounter when implementing internetworking solutions in complex IBM networks. This scenario focuses on potential implementation problems associated with translational bridging, SRT, SDLC transport, and SDLC-to-LLC translation.

Symptoms

The large-scale corporate network illustrated in Figure 2-16 is composed of multiple Ethernet and Token Ring segments partitioned with source-route bridges, SRT bridges, a transparent bridge, and a translational bridge.

Connectivity problems on this network are as follows:

1. Nonsource-route-capable end system (PC-2) on Ring #3 cannot communicate with either of the DEC LAT Servers LAT-1 and LAT-2 on Ethernet 3 and Ethernet 1, respectively.
2. Source-route-capable end system (PC-1) on Ring #3 cannot reach LAT-2 on Ethernet.
3. IBM 3174 cluster controller (Cluster-2) attached to Router-5 cannot communicate with IBM 3745 front-end processor (FEP-2) attached to Router-4.
4. IBM 3174 Cluster Controller (Cluster-1) cannot communicate with IBM AS/400 attached to Ring #2.

Environment Description

Figure 2-16 illustrates a map of the environment discussed in this case as initially configured. The following summarizes the relevant elements of this primarily bridged internetworking environment:

- The corporate network (Main-Net) consists of an Ethernet and three Token Rings separated by both Cisco and non-Cisco internetworking devices.
- Remote-Site is interconnected via a T1 serial link between Router-1 and Router-3. Remote-Site includes two Ethernets (Ethernet 2 and Ethernet 3) and a single Token Ring.
- Cisco devices are configured as follows: Router-5 is configured for SRT and STUN; Router-4 is configured for SDLC Transport; Router-3 is configured for SRT and SDLLC; Router-1 is configured for translational bridging and SRT; and Router-2 is configured for transparent bridging only.

- Non-Cisco internetworking devices at Main-Net are as follows: SRB-1 (a source-route bridge) separates Ring #1 and Ring #2; SRT-1 (a source-route transparent bridge) separates Ring #2 and Ring #3.
- Token Ring LANs are 4-Mbps and 16-Mbps, IEEE 802.5 compliant; Ethernets are IEEE 802.3 compliant.
- All the serial links from FEPs and Cluster Controllers to Cisco routers are 56-Kbps SDLC lines.
- The network applications running over the WAN include file transfer, mail, Novell, and both DEC LAT and IBM 3270 terminal connections.
- Other protocols can be routed within this environment, but the focus in this scenario is on mixed-technology bridging issues.

Remote-Site

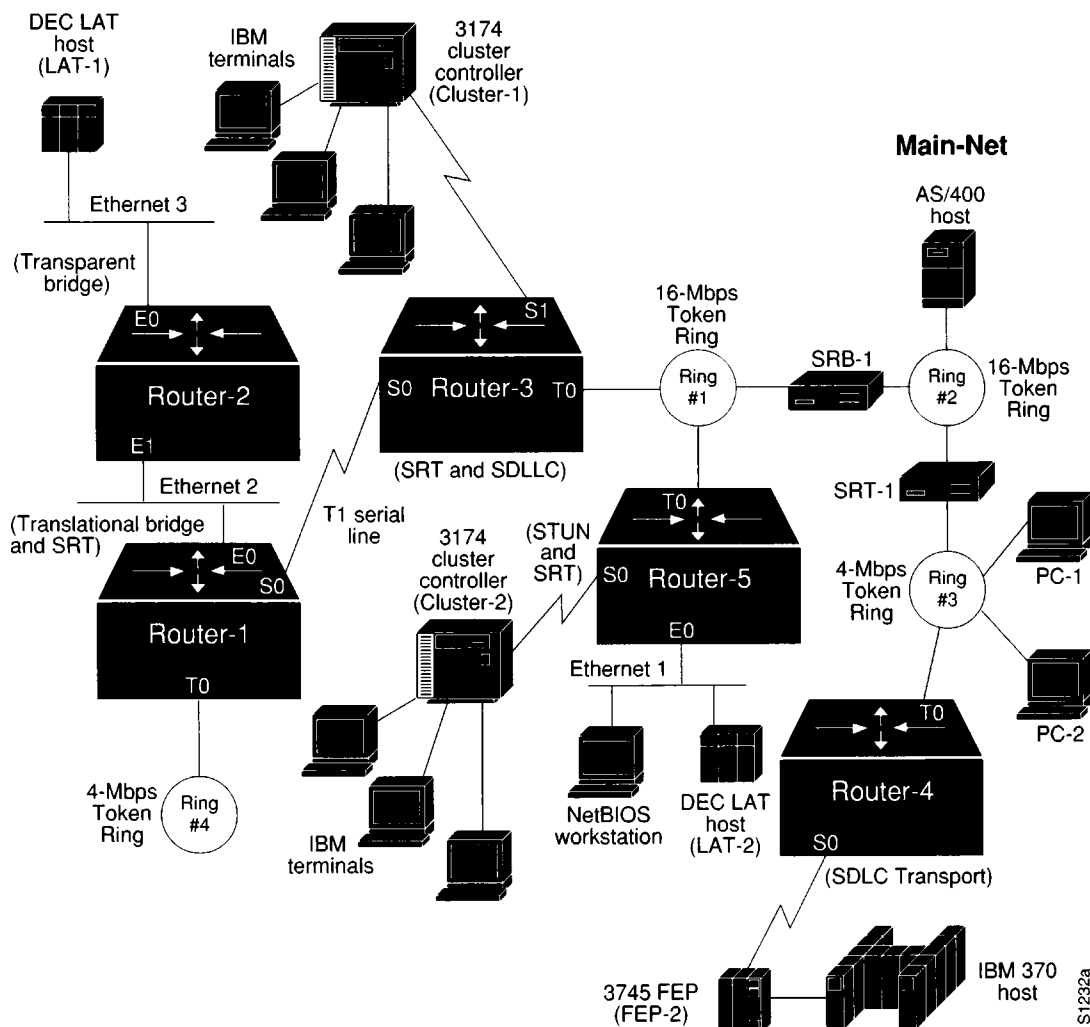


Figure 2-16 Initial IBM Internet Problem Environment

Diagnosing and Isolating Problem Causes

Before attempting to define a specific problem, it is important to identify the most likely causes and to then *systematically* eliminate each one. Given the situation, the following problems are the best candidates for interconnection failures:

- Incompatibilities between end systems and intermediate systems in mixed-media, multiprotocol environment.
- Packets with RIF being dropped by SRTs attached to Ethernets.
- Missing **ethernet-transit-oui** command.
- Missing **multiring** commands.
- Incorrectly specified addresses in router configurations.
- Missing **partner** or **sdllc xid** commands in SDLC-to-LLC translation configuration.

The next step is to eliminate each potential cause as the problem source and then test the network to determine whether it is operational. The following discussion works through the process of problem isolation.

Detecting Incompatibilities Between End Systems and Intermediate Systems

Given that the PCs on Ring #3 are trying to access DEC LAT servers on the other side of several Token Ring networks that are segmented by bridges and routers, it is possible that one or more of the PCs do not support the type of internetworking technology supported by the bridges.

In the first symptom, PC-2 is unable to access either of the target DEC LAT servers (LAT-1 and LAT-2). With an SRB in the path to both, PC-2 *itself* becomes a suspect. In particular, its ability to support SRB is in question. The following steps suggest ways to determine whether the system is source-route capable:

- Step 1:** Place a *Sniffer* on Ring #3 (same ring to which end station PC-2 is connected).
- Step 2:** Look for any frames sent by the end station (PC-2) with the high-order bit of the source address set to 1. Figure 2-17 illustrates such Sniffer output, with the high-order bit of the source address set to 1.

```

- - - - - Frame 4 - - - - -

SUMMARY  Delta T      Destination      Source      Summary
4         1.686       NetBIOS         VELA(00)    NETB Check name WWONG CISCO4

NETB: ----- NETBIOS Add Name Query -----
NETB:
NETB: Header length = 44, Data length = 0
NETB: Delimiter = EFFF (NETBIOS)
NETB: Command = 01
NETB: Response correlator = 0008
NETB: Name to be added = WWONG   CISCO4
NETB:

ADDR: HEX                                ASCII
0000  01 40 C0 00 00 00 00 80  90 00 5A DE 0D 8A C8 00  .@.....Z.....
0010  00 11 00 A1 00 20 F0 F0  03 2C 00 FF EF 01 00 00  .....
0020  00 00 00 08 00 00 00 00  00 00 00 00 00 00 00 00  .....
0030  00 00 00 00 00 57 57 4F  4E R7 20 20 20 43 49 53  ....WWONG   CIS
0040  43 4F 34 20 20                                CO4

```

Hex value of 90 is binary 1001 0000—indicating that the high-order bit of the source address is set to 1.

Figure 2-17 Sniffer Output Showing SRB-Capable End System Source Address

- Step 3:** If such a frame is not found, the end station does not support RIF and is not able to participate in source routing.
- Step 4:** To get PC-2's traffic through to LAT-1 and LAT-2, replace SRB-1 with an SRT. This network change is addressed later as part of a comprehensive solution; see "Problem Solution Summary" for a revised map and a description of the network changes involved.

Note: Make sure the client (PC-2) is configured to point to the hardware addresses for servers on Ethernet (LAT-1 or LAT-2).

Detecting SRT/SRB Incompatibilities

In Symptom #2, PC-1 (which is SRB-capable) on Ring #3 can talk to DEC LAT server LAT-1, but cannot talk to DEC LAT server LAT-2. As with the preceding problem, the key here rests with technology differences between the internetworking devices in the path to the servers and the end station trying to make a connection.

The likely stopping point for traffic in this case is Router-5, which is configured as a source route transparent (SRT) bridge. Because Router-5 is attached to both a Token Ring and an Ethernet segment (and is configured for SRT), it discards packets that include RIF data. Again, as in the prior procedure, determine whether the end system of interest (here, PC-1) is source-route capable. The steps to remedy this problem are analogous to the prior procedure, with some slight differences:

- Step 1:** Place a *Sniffer* on Ring #3 (same ring to which end station PC-1 is connected).
- Step 2:** Look for frames sent by the end station (PC-1) with the RIF present. Figure 2-18 illustrates *Sniffer* output with RIF present.

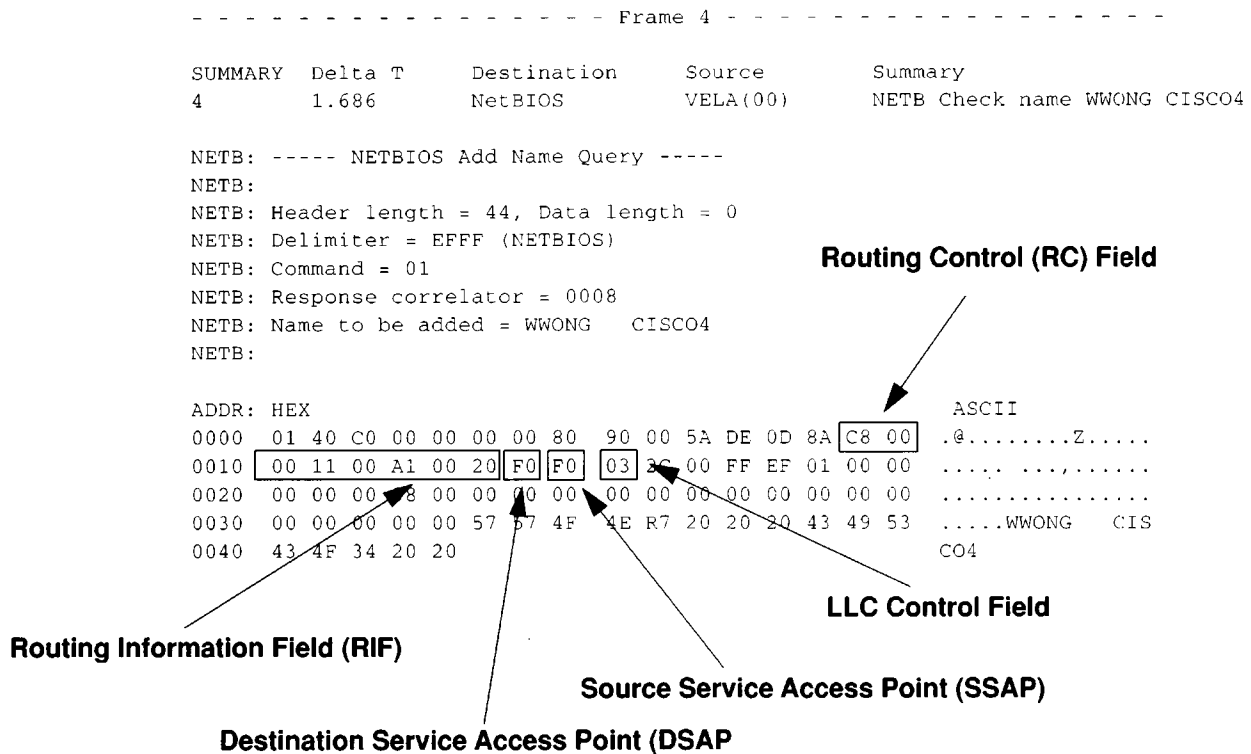


Figure 2-18 Sniffer Output Showing End System Packet with RIF

- Step 3:** If such a frame is found, PC-1 is source-route capable. The RIF illustrated in Figure 2-18 specifies that the frame came from Ring 001 (hex) over bridge 1 (hex), through Ring 00A (hex) over bridge 1 (hex) to Ring 002 (hex).
- Step 4:** In contrast to the prior problem discussion, an end station with a RIF *is a problem*. When Router-5 sees the RIF in packets sent from PC-1, it will drop those packets rather than put them on the Ethernet interface.
- Step 5:** To get PC-1's traffic through to LAT-2, you have two choices: enable translational bridging on Router-5 or replace SRB-1 with an SRT. This network change is addressed later as part of a comprehensive solution; see "Problem Solution Summary" for a revised map and a description of the network changes involved.

Note: As in the preceding procedure, make sure the client (PC-1) is configured to statically point to the hardware addresses for its servers on Ethernet (LAT-1 and LAT-2) in order to be able to listen to the service advertisements.

Resolving Vendor Code Mismatch Problems

Older Token Ring implementations, such as the IBM 8209, expect a vendor code (OUI field) of the SNAP header to be 000000. Cisco routers modify this field to be 0000F8 to specify that the frame was translated from Ethernet Version 2 to Token Ring. Cisco's modification of this field can cause end stations that expect the SNAP header to be 000000 to drop packets. Use the **ethernet-transit-oui** global configuration command to remedy this problem.

Step 1: Using the **write terminal EXEC** command, look for the **ethernet-transit-oui** global configuration command on Router-1. This command may be required on the router acting as a translational bridge.

Step 2: If frames are getting through the translational bridge, but some workstations are dropping packets (and this command is not present), specify the **ethernet-transit-oui** global configuration command on Router-1. This command forces the router to make the vendor code field 000000.

Refer to the *Router Products Configuration and Reference* publication for more information.

Finding Missing Multiring Subcommands

If routed protocols are not making it through an environment consisting of SRBs, look for missing **multiring** Token Ring interface subcommands.

Symptom #3 suggested that Cluster-2 (3174 cluster controller) cannot communicate with FEP-2. Here, SDLC transport (tunneling) is implemented via IP encapsulation. This suggests that Router-4 or Router-5 may be missing the **multiring** interface subcommand—required as a result of routing between Router-4 and Router-5. The following steps outline actions to diagnose and remedy potential configuration problems in this kind of environment.

Step 1: As an initial step, use the **ping EXEC** command to determine whether Router-5 can communicate with Router-4.

Step 2: Using the **write terminal EXEC** command (on Router-4 and Router-5), look for a **multiring** interface subcommand that includes the **ip** keyword option or the **all** keyword option (applied to the Token Ring interfaces).

Step 3: Assuming that the **multiring** command is not included or does not cover a particular protocol that is being routed (and subsequently bridged over the SRB as in this scenario), one alternative is to add the **multiring ip** command to Router-4 (Token Ring interface T0) and Router-5 (Token Ring interface T0), as illustrated in the initial network map (Figure 2-16). Refer to the *Router Products Configuration and Reference* publication for more information about using the **multiring** command.

Step 4: Another option is to reconfigure the network so that this problem is eliminated. Refer to Figure 2-19 in the “Problem Solution Summary” discussion to see how this can be done. In this case, removing SRB-1 and SRT-1 remedies the problem without requiring the addition of the **multiring ip** command.

Enabling Access to the AS/400 on Ring #2

The last symptom in our scenario indicates that Cluster-1 cannot talk to the AS/400 directly attached to Ring #2.

The following steps outline actions to isolate the reason that connections from Cluster-1 to AS/400 are blocked and to then enable access.

- Step 1:** Place a network analyzer on Ring #1 (same ring to which Router-3 is connected).
- Step 2:** Determine whether Router-3 is generating explorer packets.
- Step 3:** If Router-3 is not generating any explorer packets for the AS/400, check its configuration for inclusion of the **partner** global command and **sdlc xid** interface subcommand.
- Step 4:** If not present, add the **partner** and **sdlc xid** commands. These additions force the router to generate explorer packets.

Problem Solution Summary

As indicated early in this case, several of the solutions pointed to a redesign of the original network as illustrated in Figure 2-16. Figure 2-19 illustrates a suggested reconfiguration of the internetwork. The modification is the use of a Cisco router (here an AGS+) to replace SRB-1 and SRT-1, and the implementation of SRT on all Main-Net Token Ring links.

This scenario addressed a number of common problems encountered in complex IBM internets. Actions discussed included:

- Resolving SRB-related and SRB/SRT technology conflicts by designing out SRT-1 and SRB-1 and replacing them with an AGS+ router (Router-4).
- Using third-party diagnostic tools to isolate problems based on traffic occurring on a network.
- Adding missing **ethernet-transit-oui** command to applicable configurations to resolve vendor code mismatch problems (Router-1, global configuration change).
- Fixing incorrectly specified SDLC address of Cluster-2 in router configurations (Router-3, interface Serial1, and Router-5, interface Serial0).
- Adding missing **partner** commands in SDLC-to-LLC translation configurations (Router-3, interface Serial1).

Figure 2-20 through Figure 2-23 provide the complete, final configuration listings for the key routers discussed in this scenario, illustrating configurations required to interconnect this internet.

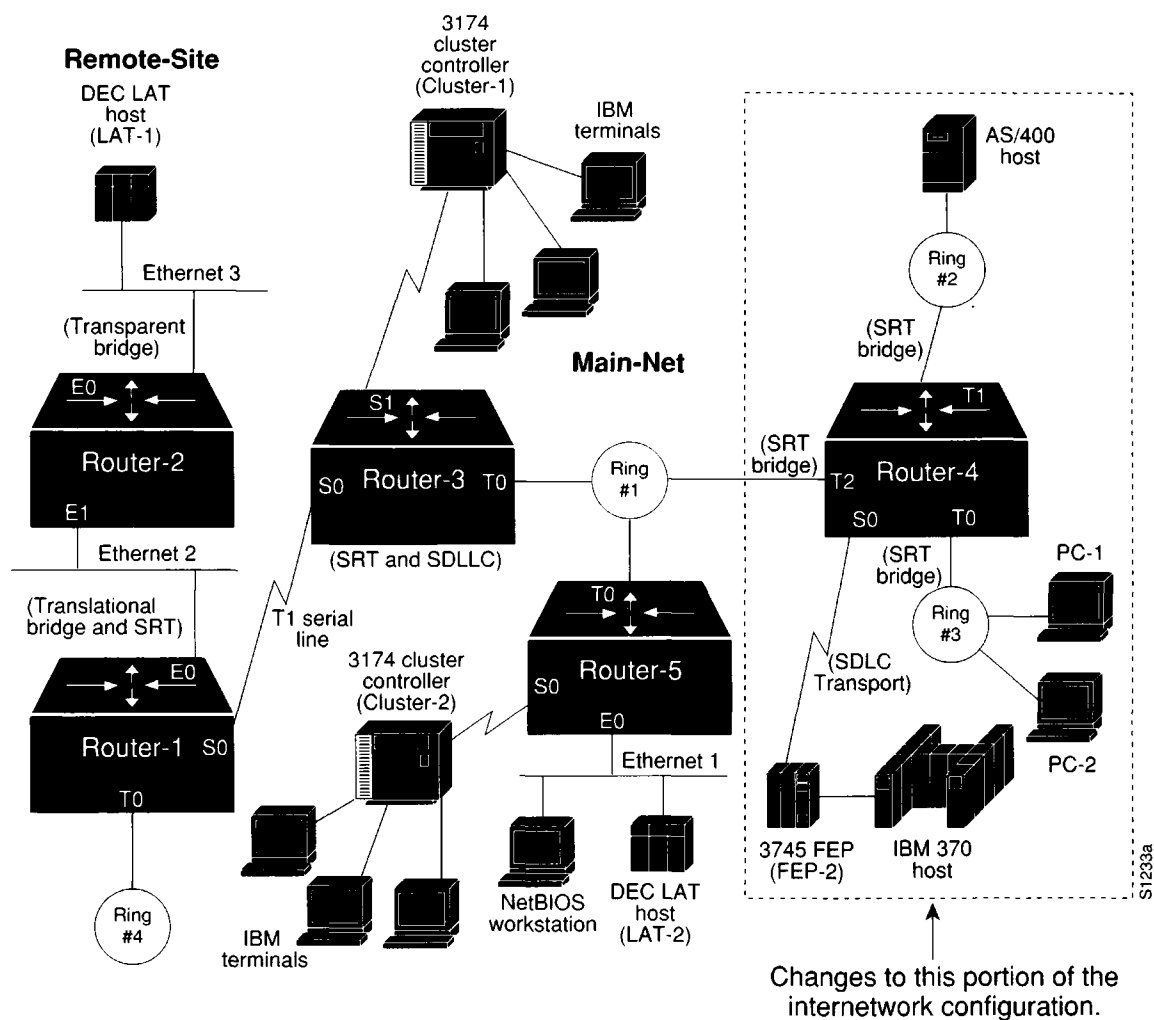


Figure 2-19 Reconfigured IBM Internet Environment

```

!
!
source-bridge ring-group 10
source-bridge remote-peer 10 tcp 131.108.1.1
source-bridge remote-peer 10 tcp 131.108.1.2
source-bridge remote-peer 10 tcp 131.108.2.2
source-bridge transparent 10 5 1 1
!
!
interface TokenRing 0
no ip address
ring-speed 16
source-bridge 4 1 10
source-bridge spanning
multiring all
bridge-group 1
!
!
interface Ethernet 0
no ip address
bridge-group 1
!
!
interface Serial 0
ip address 131.108.1.1 255.255.255.0
bridge-group 1
!
!
router igrp 109
network 131.108.0.0
!
ip name-server 255.255.255.255
snmp-server community
bridge 1 protocol ieee
!
ethernet-transit-oui standard

```

Figure 2-20 Relevant Router-1 Final Configuration Listing

```

!
!
source-bridge ring-group 10
source-bridge remote-peer 10 tcp 131.108.1.1
source-bridge remote-peer 10 tcp 131.108.1.2
!
!
!
interface TokenRing 0
no ip address
ring-speed 16
source-bridge 1 1 10
source-bridge spanning
bridge-group 1
!
!
interface Serial 0
ip address 131.108.1.2 255.255.255.0
bridge-group 1
!
!
interface Serial 1
no ip address
encapsulation sdhc-primary
sdhc address c1
sdllc traddr 0110.2222.3300 8 1 10
sdllc partner 0000.2000.0400 c1
!
!
router igrp 109
network 131.108.0.0
!
ip name-server 255.255.255.255
snmp-server community
bridge 1 protocol ieee
!

```

Figure 2-21 Relevant Router-3 Final Configuration Listing

```

stun peer-name 131.108.2.2
stun protocol-group 1 sdic
!
!
source-bridge ring-group 10
source-bridge remote-peer 10 tcp 131.108.1.1
source-bridge remote-peer 10 tcp 131.108.2.2
!
!
interface TokenRing 0
no ip address
ring-speed 16
source-bridge 3 1 10
source-bridge spanning
multiring all
bridge-group 1
!
interface TokenRing 1
no ip address
ring-speed 16
source-bridge 2 1 10
source-bridge spanning
bridge-group 1
!
!interface TokenRing 2
ip address 131.108.2.2 255.255.255.0
ring-speed 16
source-bridge 1 2 10
source-bridge spanning
bridge-group 1
!
interface Serial 0
encapsulation stun
no ip address
no keepalive
stun group 1
stun route address c2 tcp 131.108.2.3
!
!
router igrp 109
network 131.108.0.0
!
ip name-server 255.255.255.255
snmp-server community
bridge 1 protocol ieee
!

```

Figure 2-22 Relevant Router-4 Final Configuration Listing

```

stun peer-name 131.108.2.3
stun protocol-group 1 sdlc
!
source-bridge ring-group 10
!
!
interface TokenRing 0
ip address 131.108.2.3 255.255.255.0
ring-speed 16
bridge-group 1
!
interface Ethernet 0
no ip address
bridge-group 1
!
interface Serial 0
encapsulation stun
no ip address
no keepalive
stun group 1
stun route address c2 tcp 131.108.2.2
!
!
ip name-server 255.255.255.255
snmp-server community
bridge 1 protocol ieee

```

Figure 2-23 Relevant Router-5 Final Configuration Listing

Novell Network Server Connectivity Scenario

With the emergence of Novell NetWare as the dominant PC-based network operating environment, network administrators have encountered increasing requirements to interconnect and segment PC LANs running Novell's IPX networking protocol. This scenario focuses on a variety of problems that can impair server access over a routed internet.

Symptoms

Figure 2-24 is a map of the Novell IPX internet discussed in this case. It illustrates an interconnection between two sites over an arbitrary serial network. The following facts summarize the situation:

- Client-A cannot access Server-1 and Server-2 on the other side of the serial link. However, Client-A can access Server-3 on the local wire.
- Client-N (a NetBIOS client) cannot access Server-N (a NetBIOS-based CD-ROM server), which is also on the other side of the link.

Since no connections can be made over the serial link, it *initially* appears that there is a problem with traffic getting through the routers.

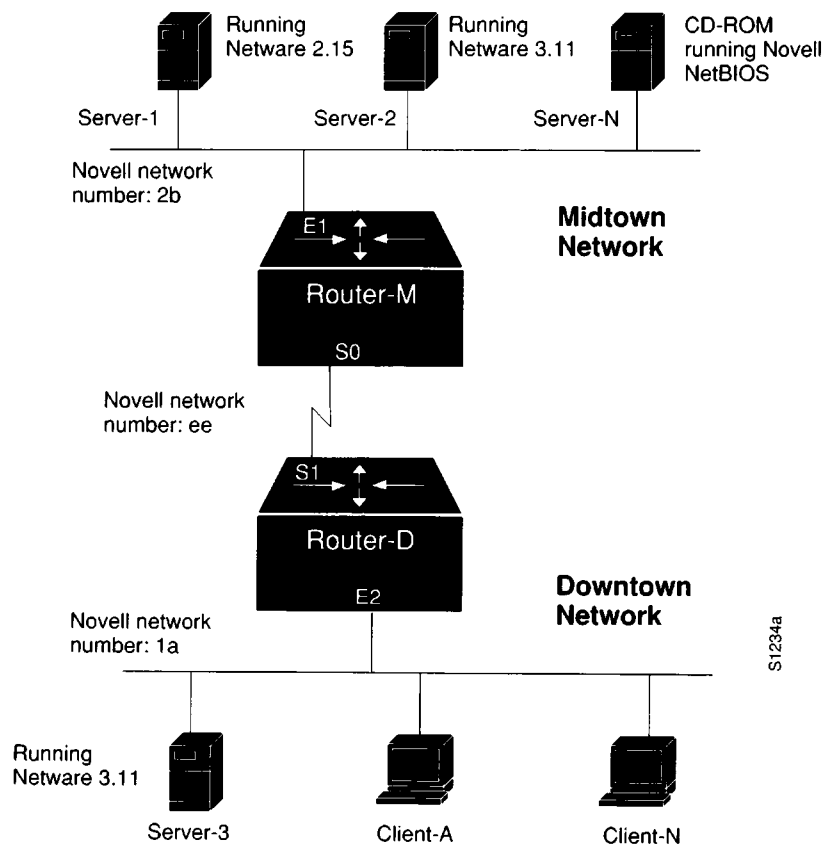


Figure 2-24 Initial Novell IPX Connectivity Scenario Map

Environment Description

The relevant elements of the internetworking environment shown in Figure 2-24 can be summarized as follows:

- Remote service is provided to a cross-town campus via a point-to-point serial link.
- Two routers (Router-M and Router-D) interconnect the Midtown and Downtown networks. Routers are MGS routers, configured to route IPX. The clients are IBM PCs and compatibles.
- The LANs are Ethernets; the serial link is a dedicated T1 link (1.544 Mbps).
- The network applications intended to run over the T1 line include typical NetWare services.
- Server-1 is running NetWare 2.15, while Server-2 and Server-3 are running NetWare 3.11. Server-N is a CD-ROM running Novell NetBIOS.

Diagnosing and Isolating Problem Causes

Given the situation, a number of problems could explain both connectivity symptoms.

The following problems are likely candidates for the first symptom (Client-A cannot access services on Server-1 and Server-2):

- Client-A and/or target servers are not properly attached to their networks
- Novell routing is not enabled on Router-D and Router-M
- Misconfigured network numbers
- Router interfaces are not up or operational
- Server-1 and Server-2 are running limited-user versions of NetWare
- Server-2 is specifically *not* configured to send SAP updates
- Encapsulation type mismatch
- Nonunique MAC addresses in Novell routing configuration
- Misconfigured access list
- Backdoor bridge

The following problems are likely candidates for the second symptom (Client-N cannot access services on NetBIOS server):

- Client-N and/or target server are not properly attached to their networks
- Misconfigured network numbers
- Novell routing is not enabled on Router-D and Router-M
- Router interfaces are not up or operational
- Server-N is running a limited-user version of NetWare
- Encapsulation type mismatch
- Nonunique MAC addresses in Novell routing configuration

- Misconfigured access list
- Bad or missing helper address
- Backdoor bridge

Both lists are loosely ordered according to a combination of two criteria: ease of problem determination and likelihood of being the *actual* problem.

The problems identified as likely to block service access for Client-A and Client-N are essentially the same, with slight variations. In general, it is useful to eliminate most-likely problems first, and then to tackle more complex problems as necessary. The problem-solving process that follows illustrates this strategy.

Once a possible problem list is determined, you must analyze each potential cause. The following discussion considers the problems listed and illustrates resolution of discovered problems.

Checking Physical Attachment of Clients to Network

The first step is to determine whether Client-A is attached to the network. This step applies for Client-N as well, and can be done at the same time.

- Step 1:** Visually inspect each client's physical attachment and attempt to connect to a local server. If a connection can be established, the client is obviously attached.
- Step 2:** If a connection cannot be established to a local server (either one does not exist or the connection attempt fails), use a protocol analyzer to determine whether clients are sending any packets. Look for packets with each client's hardware address as the source address.
- Step 3:** As an alternative, use the **debug novell-packet** EXEC command on the locally connected router (in this case Router-D) and look for each client's source address. If packets appear that include the respective client's hardware address as the source address, the client is active on the network and connectivity to Router-D is functional.

In order to use **debug novell-packet**, you must disable fast switching (use the **no novell route-cache** command on Ethernet interface E2). In addition, if you are making a virtual connection to the router, be sure to use the **terminal monitor** EXEC command so the **debug** command output will appear on your remote terminal.

Note: You can also use the the Novell server command **track on** to determine whether servers are broadcasting. Simple client/server activity can be viewed in this fashion. The **track on** command is a server console command.

In this case, assume that connectivity is verified up to Router-D from both Client-A and Client-N.

Checking Physical Attachment of Servers to Network

The next step is to determine whether the remote servers are attached to their Ethernet segment. This process is very similar to determining whether the clients are attached to the Downtown segment. However, there are some slight differences.

- Step 1:** As in the prior step, start by visually inspecting the attachment of the servers to their networks.
- Step 2:** Using a protocol analyzer, determine whether the servers (in this case Server-1, Server-2, and Server-N) are sending any packets on their local networks. Look for packets with each server's hardware address as the source address.
- Step 3:** Check for connectivity between the servers and Router-M. To do this, use **show novell servers** to see if the servers are included in the router's list of Novell servers. If they appear in the list, then connectivity is verified up to Router-M.

In this case, assume that connectivity is verified up to Router-M from both Server-1 and Server-N; however, Server-2 does not appear in the **show novell servers** output for Router-M.

Before continuing, you must determine why Server-2 is not appearing in the Novell server list on Router-M.

Enabling Novell IPX Routing

Use the **write terminal EXEC** command to determine whether Novell routing is enabled. Use the **novell routing** global configuration command if it is not.

Determining Whether SAP Updates Are Disabled

Although not necessarily an obvious problem, the fact that Server-2 is not appearing in the server table for Router-M suggests that there is a configuration problem with Server-2.

By default, Novell servers send SAP updates to tell clients what services are available. However, servers running NetWare 3.11 or later can be specifically set to withhold SAP updates. If a server is set to withhold SAP updates, the local router does not broadcast SAP updates across the serial link, and clients cannot access the server's services.

- Step 1:** To determine whether a server is configured to withhold SAP updates, you must examine the specific server's configuration. Read the server's documentation to determine how to find this information.
- Step 2:** Assume that Server-2 was set to withhold SAP updates. Change this configuration.
- Step 3:** Again, check for connectivity between Server-2 and Router-M, using the **show novell servers** command. Assume that Server-2 appears.

Unfortunately, Client-A is still unable to access Server-1 and Server-2, while Client-N is still unable to access the NetBIOS server (Server-N).

Checking Novell Network Number Specifications

Next, examine the network number specifications for servers and routers on all networks in the internet.

Step 1: Assuming that routing is enabled, compare the specifications for the Novell network number (using the **novell network number** command) on each router interface.

Step 2: Look for missing or duplicate network number specifications. If you find duplicates, assign unique network numbers for each network segment.

In this case, assume there is a subtle conflict. The network number assigned for the serial link is "ee." Unfortunately, this is also the internal network number assigned to Server-3. The result is that there is no connectivity over the serial line between Midtown and Downtown. The solution is to modify the serial line network number to something else (for example, "af"). Figure 2-25 illustrates this change.

However, when this change is made, there is no change to service availability.

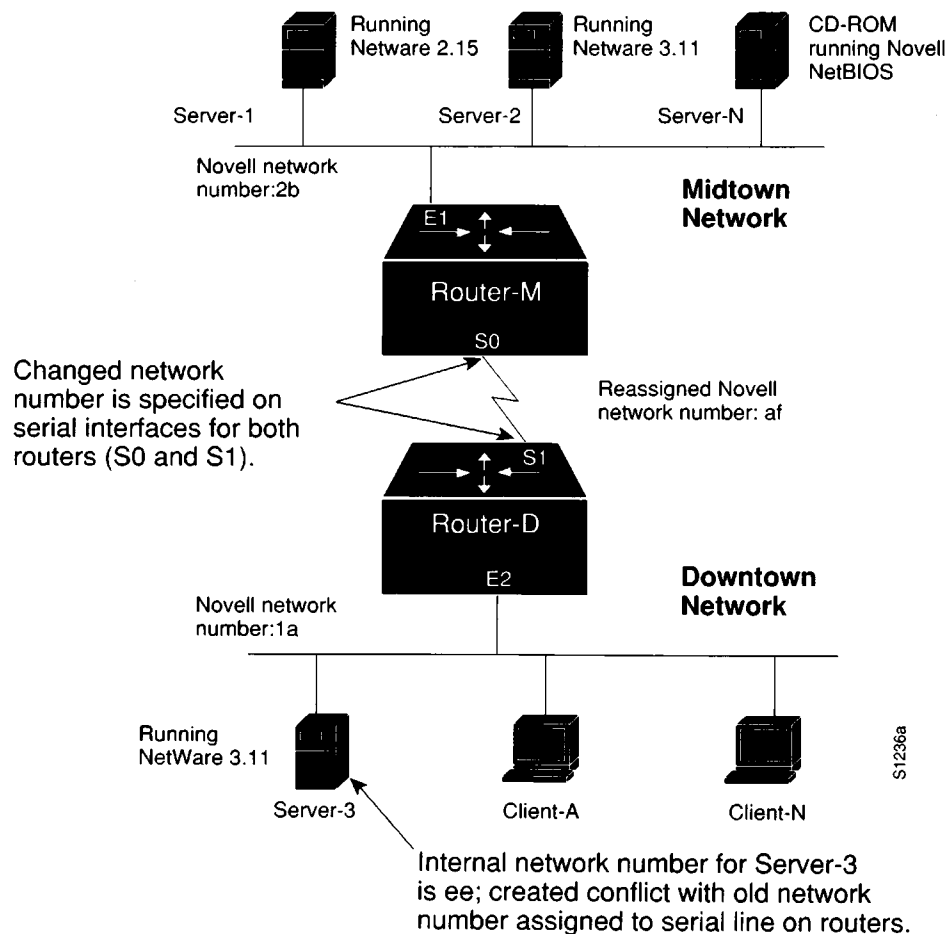


Figure 2-25 IPX Connectivity Map Showing Revised Network Number

Checking Router Interface Status

In the process of eliminating the preceding problems, it is highly likely that the status of each router interface has been verified.

- Step 1:** You can further confirm the status of the router interfaces using the **show novell interface** command on each router. The interfaces should indicate that the interface is up and that the protocol is up.
- Step 2:** You can also **ping** between the routers to confirm that the interfaces are operational.

Again, for the purposes of this case, assume that the interfaces are all functional.

Checking for Limited-User Version of NetWare

In some cases, NetWare server software may limit the number of users that can access the server simultaneously. If your copy is a limited-user version, you must upgrade the version to support more users.

In this case, the version can be assumed to be a standard version supporting more users. Client-A is still unable to access Server-1 and Server-2, and Client-N is still unable to access Server-N.

Checking for Encapsulation Mismatch

Next on the problem list is an encapsulation mismatch. The default on Cisco routers is Novell's Frame Type Ethernet_802.3 encapsulation. If there is a conflict (any entity is configured for a different framing from the rest of the internetwork's entities), you must modify all configurations so that they match.

- Step 1:** To determine what framing type clients (and servers) are running, you might change the framing type on the local router (Router-D for the clients and Router-M for the servers) to **arpa** (Novell's Frame Type Ethernet_II).
- Step 2:** Next, enable **debug novell-packet** on the local router. If you see a packet with the source address of a client or server, that node is using Frame Type Ethernet_II. (Remember to disable fast switching using the **no novell route-cache** command before enabling this **debug** command.)
- Step 3:** You also can use the **show novell traffic EXEC** command to look for an incrementing "format errors" counter. This suggests that there is an encapsulation mismatch.
- Step 4:** As an alternative to using these Cisco-specific commands, you can use a protocol analyzer to capture packets. Examine packets from clients, servers, and routers and determine whether they are all using the same framing type. If not, change configurations on nodes so that all are using the same encapsulation type.

Note: Different encapsulation types can coexist on the same wire and in the same inter-network; however, Cisco routers can translate between encapsulation types on the same segment only when *more than one* interface is attached to that segment. If you require that Frame Type Ethernet_II and Ethernet_802.3 both be supported simultaneously, you must have two separate interfaces attached to the same network segment—with each supporting different framing types. (Note that each interface must use a different network number.) In addition, Cisco routers currently do not support SNAP encapsulation over Ethernet.

In this case, assume that all nodes are using Frame Type Ethernet_802.3 (the Cisco router default).

Checking for Access List Problems

Access list problems come up in many connectivity problem lists. For details concerning access list issues, refer to Chapter 5, “Troubleshooting Novell Connectivity.”

For the purposes of this case, assume that the **write terminal** output for both Router-D and Router-M indicates that there are no relevant access list specifications.

Checking for Nonunique MAC Addresses on Routers

MAC addresses are obtained for Novell configurations in one of two ways: either from the router hardware address embedded in the system firmware or by random assignment (when the system software initializes before the interface is initialized). In some *rare* cases, the randomly generated MAC address for different routers will be the same. If these numbers are not unique, and the routers are on the same internet, communication will not occur. If Router-M and Router-D have the same MAC address, no traffic will traverse the serial link.

- Step 1:** To determine if this is the problem, use the **write terminal** command to examine the current configuration of each router in the path (Router-D and Router-M).
- Step 2:** Check the hardware address specified in the **novell routing** global configuration command. If this system-generated number matches for both routers, reinitialize one of the routers and see if connectivity over the link is re-established.
- Step 3:** Test for connectivity between clients and servers.
- Step 4:** If connectivity is still blocked, reexamine the configuration of the routers.
- Step 5:** If the routers still have matching MAC addresses, use the **show controllers interface-type** command to obtain an actual MAC address from each router.
- Step 6:** Use the **novell routing** command to enter the selected MAC address (for example, **novell routing 00aa.54f1.003e**).

In general, this problem occurs more frequently in Token Ring implementations. For the purposes of this case, assume that the addresses are different.

Checking for Misconfigured Helper Addresses

Next, look for a missing or misconfigured Novell helper address specification on one of the routers.

Step 1: To diagnose this configuration problem, use the **write terminal EXEC** command to look for **novell helper-address** interface subcommand specifications. This command must include either an *all nets* specification (-1.ffff.ffff.ffff) or *directed broadcast* specification (2b.ffff.ffff.ffff).

Note: If the all nets specification is used, it must be specified on Router-M (serial interface S0) and Router-D (Ethernet interface E2). Figure 2-26 illustrates the flow of broadcast traffic from clients and the application of the all nets broadcast specification. If a directed broadcast specification is used, it is only required on Router-D (Ethernet interface E2).

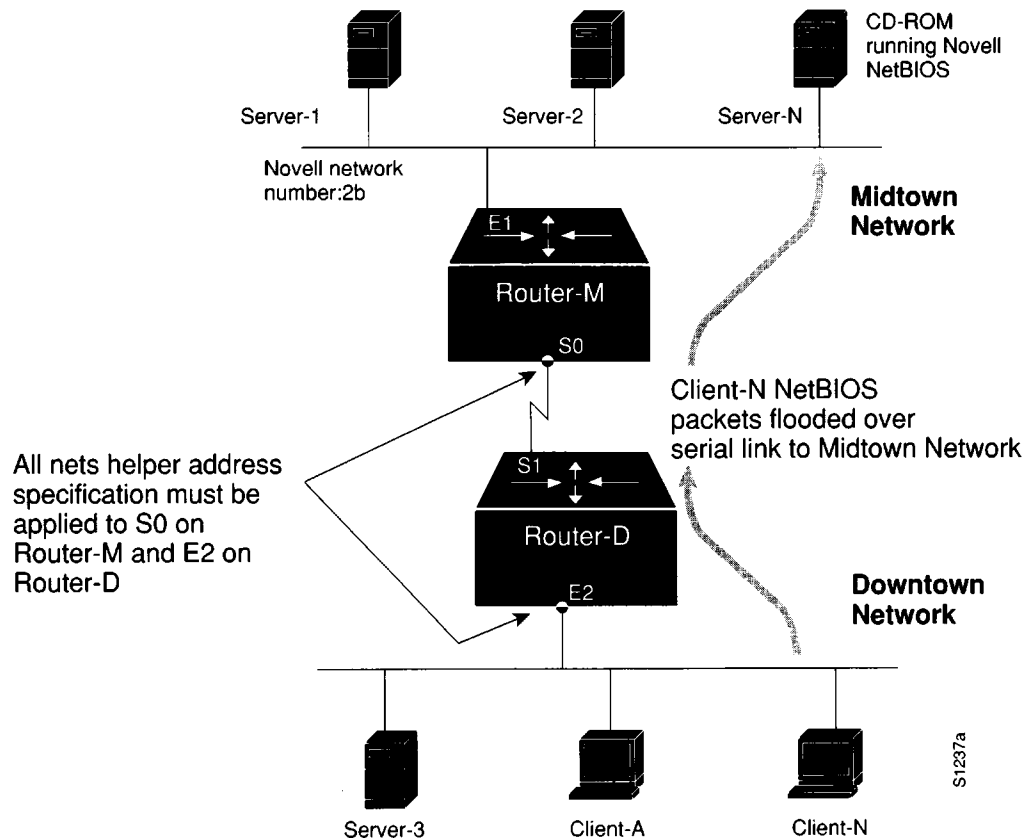


Figure 2-26 All Nets Helper Address Specification Illustration

Assume that the helper address is not included in the original configuration and is added as a correction. Unfortunately, connectivity to the remote hosts from Client-A and Client-N is still blocked when test connections are attempted.

Another helper address-related problem might be that a NetBIOS server is using reverse broadcast to communicate with clients, but the routers are not configured to accommodate this requirement. In this case, assume that the servers are not attempting to reverse broadcast to clients.

Finding a Backdoor Bridge

Finding a back door in your internetwork can be an extremely difficult task. A backdoor bridge typically results from the process of migrating from a bridged internetwork to a routed environment. For one reason or another, bridges sometimes are not removed. Unfortunately, if backdoor bridges exist, route information leaks between the networks and routers are defeated. The result is a complete blockage of connectivity between segments.

Step 1: To find a back door, use a protocol analyzer to look for packet loops. In particular, look for routing and SAP updates from remote networks.

Also, look for known *remote* network numbers that show up on the *local* network. That is, look for remote network numbers that are not being handled by the router (source address is not the router's).

Step 2: Watch for changes in hop counts. If a back door exists, you are likely to see hop counts incrementing up to 16; routes may disappear and then reappear unpredictably.

Step 3: If you observe any of these situations, isolate the local Ethernet into smaller segments (using a fanout or similar device).

Step 4: Examine the traffic on each segment with a protocol analyzer. If a packet appears from a known remote node (has a remote source address), the back door is located on that segment.

If a backdoor bridge is discovered (as illustrated in Figure 2-27), you must remove the link. This link might be inadvertently configured within a wiring concentrator or might be two bridges anywhere in your network.

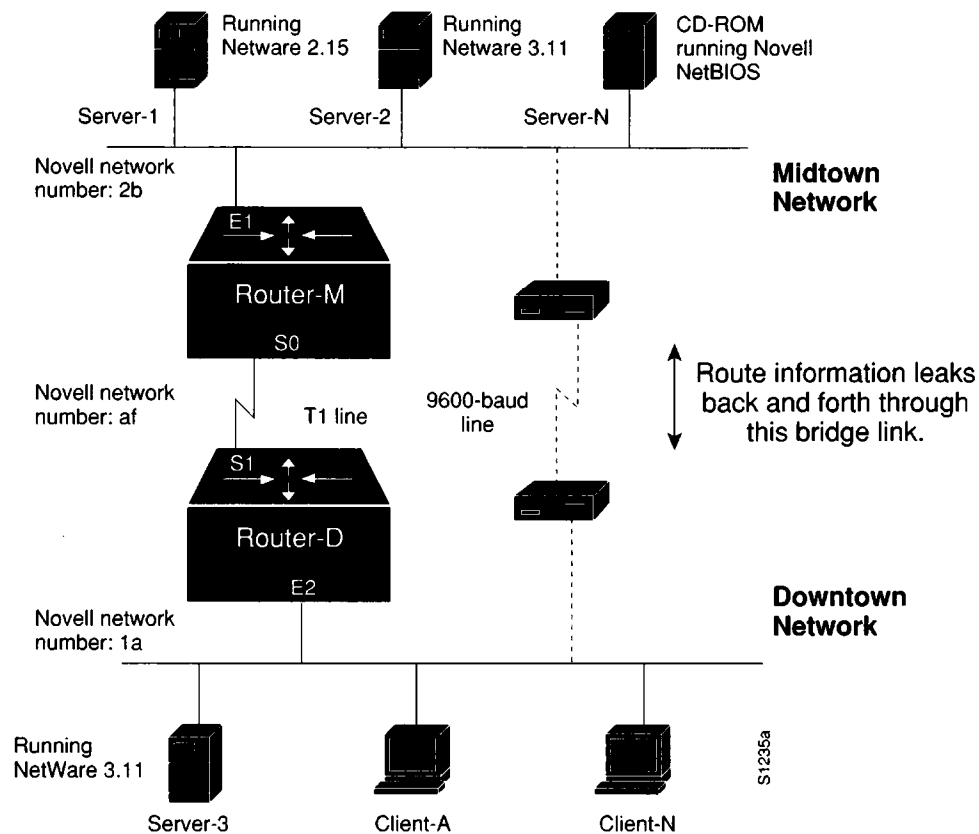


Figure 2-27 Novell IPX Connectivity Scenario Map with Backdoor Bridge Shown

Removing this interconnection finally establishes connectivity between the two segments. Client-A and Client-N are now able to access the Midtown servers.

Problem Solution Summary

This scenario focused on diagnosing blocked connectivity in Novell IPX internets. Four problems were discovered and resolved:

- Server-2 was reconfigured to send SAP updates.
- Misconfigured network numbers were corrected.
- A directed-broadcast helper addresses was included to broadcast Novell NetBIOS client requests.
- An unexpected back door was removed, which ultimately allowed connectivity between the segments.

Figure 2-28 and Figure 2-29 provide representative configuration listings for Router-M and Router-D as discussed in this scenario. These configurations illustrate the configuration commands required to interconnect the two Ethernet segments over the T1 line. (Remember to reenale fast switching using the **novell route-cache** command if it was disabled during troubleshooting.)

```
novell routing
!
!
interface Ethernet 2
novell network 1a
novell helper-address 2b.ffff.ffff.ffff
!
interface Serial 1
novell network af
!
```

Directed broadcast specification




Figure 2-28 Relevant Router-D Configuration Commands

```
novell routing
!
!
interface Ethernet 1
novell network 2b
!
interface Serial 0
novell network af
!
```

Figure 2-29 Relevant Router-M Configuration Commands

TCP/IP Route Redistribution and Access Control Scenario

Many of the world's largest internetworks employ TCP/IP as their backbone network protocol. However, that does not mean that these networks employ universal internetworking implementations. In fact, TCP/IP internetworks—sometimes comprising thousands of internetworking nodes—can span organizational domains that employ completely different topologies, routing protocols, and possibly conflicting administrative objectives. The challenge is to provide the requisite level of connectivity between hosts in different domains and on different major networks, while providing adequate security for each organization attached to the internetwork. This scenario focuses on the issue of balancing connectivity and security.

Symptoms

This scenario addresses connectivity problems in TCP/IP internetworks. Figure 2-30 illustrates an organization's interconnections from one subnet to its corporate network and interconnections to external networks.

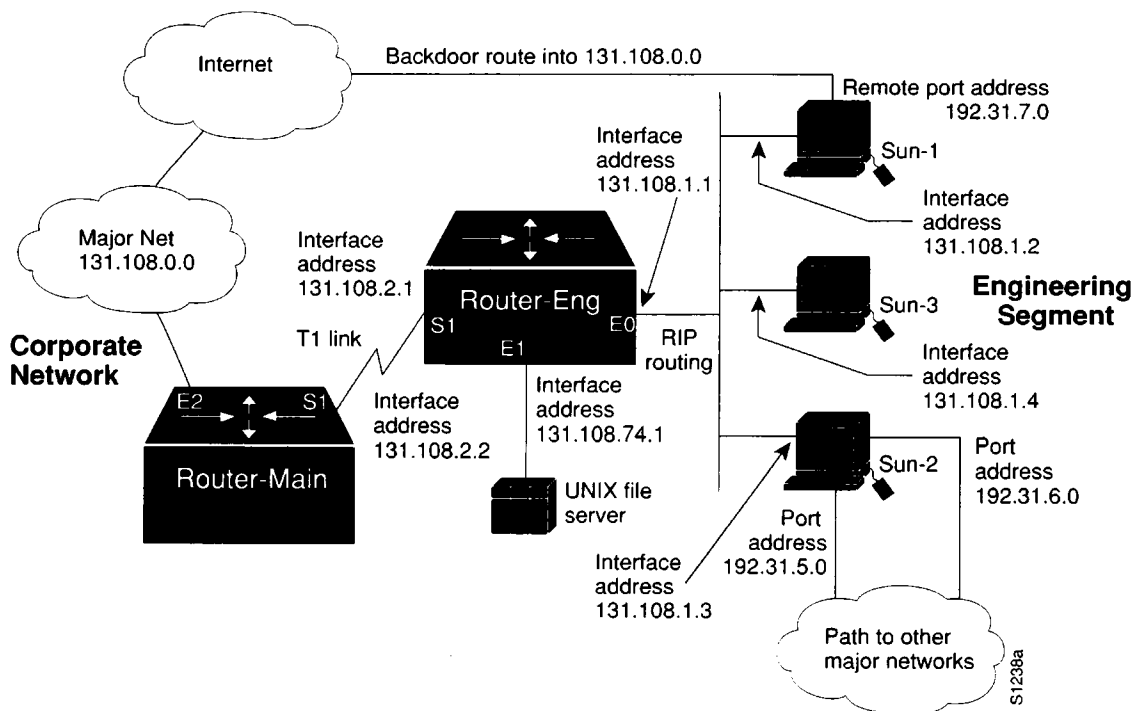


Figure 2-30 TCP/IP Internetwork Connectivity Scenario Map

Sun-1, Sun-2, and Sun-3 on the Ethernet segment attached to Router-Eng are unable to communicate with hosts in the main corporate network or outside the organization through Router-Eng. Several backdoor routes also exist, which allow other networks to access the Engineering Segment.

Because external access is not being reliably controlled, and users on the engineering segment are unable to get through to the corporate network via Router-Eng, this is both a security problem and a connectivity problem.

Environment Description

The relevant elements of the internetworking environment shown in Figure 2-30 can be summarized as follows:

- Remote service is provided to a geographically separated network via a point-to-point serial link.
- Two routers (Router-Main and Router-Eng) interconnect the engineering segment with the corporate network.
- The corporate network is interconnected to a large internetwork.
- Several backdoor routes into the engineering segment are available through serial connections to two of the UNIX hosts.
- The LANs are all IEEE 802.3 Ethernet; the serial link from the engineering segment to the corporate network is a dedicated T1 link (1.544 Mbps). The backdoor links to the UNIX workstation-based routers are asynchronous lines.
- The only network layer protocol running in this network is IP; the engineering segment is using RIP locally. The corporate network uses IGRP.
- The network applications intended to run over the T1 line are limited to file transfer (FTP), mail (SMTP), and virtual terminal connections (Telnet).

Diagnosing and Isolating Problem Causes

Given the situation, the following candidates are likely causes of this network's interconnection problems:

- Misconfigured route redistribution
- Misconfigured access lists

The next step is to analyze each potential cause as the problem source and then test the network to determine whether it is operational after any modifications are made. The following discussion considers these possible problems and alternatives for providing the proper access and security.

Isolating Router Software Configuration Problems

Because the UNIX workstation-based routers on the engineering segment are using RIP to route among themselves, while the corporate network uses IGRP, the first configuration issue to consider is route redistribution.

Step 1: Use the **write terminal EXEC** command to review the configuration on Router-Eng. In order for RIP routes and IGRP routes to be passed between the

engineering segment and the corporate network, Router-Eng must be configured for redistribution.

Step 2: Assuming that Router-Eng does not have redistribution configured, add appropriate redistribution commands.

Figure 2-31 illustrates a partial configuration for Router-Eng that would establish RIP-IGRP route redistribution for this network.

```
router rip
distance 255
network 131.108.0.0
passive-interface serial 1
default-metric 2
redistribute igrp 101
!
router igrp 101
network 131.108.0.0
passive-interface ethernet0
!
```

Figure 2-31 Example RIP-to-IGRP Route Redistribution Configuration

Note the following points about this brief example:

- The use of the **passive-interface** command prevents RIP from running on the serial network (Serial1) and blocks IGRP from running on Ethernet0.
- The **default-metric** value is assigned for the redistribution of IGRP routes sent into the RIP domain.

Step 3: At this point, you might perform an extended **ping** from Router-Main to one or more of the UNIX nodes on the engineering segment. Assuming that no access controls are in place, the **ping** should be successful and now Sun-1, Sun-2, and Sun-3 should be able to communicate with the corporate network resources.

However, setting up redistribution does not provide any means of blocking the uncontrolled backdoor access available through the asynchronous lines on the UNIX routers (Sun-1 and Sun-2).

Step 4: The next step is to set up access lists to allow Sun-1, Sun-2, and Sun-3 on the engineering segment to access the corporate network, but to block access from outside the corporation to resources on the corporate network.

Step 5: Figure 2-32 illustrates example additions for Router-Eng to control access to the corporate network.

```
interface serial 1
ip access-group 20
!
access-list 20 permit 131.108.1.2
access-list 20 permit 131.108.1.3
access-list 20 permit 131.108.1.4
```

Figure 2-32 Access Control Additions to Router-Eng Configuration

These access list entries provide the following controls:

- Access list 20 and the **ip access-group 20** interface subcommand (applied to Serial1), permit Sun-1, Sun-2, and Sun-3 on Ethernet0 to make connections *through* Serial1. However, other access via Serial1 is blocked.
- Figure 2-33 illustrates a modification to the access list specification for Router-Eng that provides a slightly different access control. Access list 21 also illustrates how order can be crucial in access list specifications. Here, the first line of access list 21 specifies that if the packet comes from address 131.108.1.4, it will be blocked (denied). If the packet is not from this source address, the next line is read. This line indicates that any packets from any other node on subnet 131.108.1.0 are permitted out Serial1—specifically packets from 131.108.1.2 and 131.108.1.3.
- If the **permit** and **deny** statements for access list 21 are swapped, all packets on subnet 131.108.1.0 are permitted. The second line is never applied, because 131.108.1.4 has *already passed* the first list entry. All other traffic is denied.

```
interface serial 1
ip access-group 21
!
access-list 21 deny 131.108.1.4
access-list 21 permit 131.108.1.0 0.0.0.255
```

Figure 2-33 Standard Access Control for Router-Eng Configuration

- Another access list variation is an extended access list. Figure 2-34 illustrates an extended access list that is used to limit access to resources by Sun-1 and Sun-2. This access lists uses source and destination filtering to control traffic from the UNIX nodes on Ethernet0. As specified, Sun-1 and Sun-2 can only access resources directly connected to 131.108.0.0. Traffic intended for any other network will not be allowed out Serial1.

```
interface serial 1
ip access-group 101
!
access-list 101 permit ip 131.108.1.2 0.0.0.0 131.108.0.0 0.0.255.255
access-list 101 permit ip 131.108.1.3 0.0.0.0 131.108.0.0 0.0.255.255
```

Figure 2-34 Extended Access Control for Router-Eng Configuration

Problem Solution Summary

This scenario focused on solving two problems in TCP/IP internets:

- Allowing the proper redistribution of routing information between different domains
- Providing appropriate access to network resources while establishing controls that limit access to networks from external hosts.

Of these two, implementing redistribution is relatively straightforward, while access lists can be fairly complicated and can yield unpredictable results.

Figure 2-35 illustrates a complete router configuration for Router-Eng (obtained with the **write terminal** command).

```

Current configuration:
!
enable-password noBuGZ
!
boot host Router-Eng-config 131.108.2.20
boot system gs3-bf.shell 131.108.2.20
!
interface Ethernet 0
ip address 130.108.1.1 255.255.255.0
!
interface Ethernet 1
ip address 130.108.74.1 255.255.255.0
!
!
interface Serial 1
ip address 131.108.2.1 255.255.255.0
ip access-group 20
!
router rip
default-metric 2
network 131.108.0.0
distance 255
redistribute igrp 101
passive-interface Serial 1
!
router igrp 101
network 131.108.0.0
passive-interface Ethernet 0
!
!
ip domain-name cisco.com
ip name-server 255.255.255.255
snmp-server community
snmp-server community dink RO
snmp-server host 131.108.2.30 dink
access-list 20 permit 131.108.1.4
access-list 20 permit 131.108.1.2
access-list 20 permit 131.108.1.3
hostname Router-Eng
!
!
line vty 0 4
login
line con 0
exec-timeout 0 0
password nErdKnoBs
line aux 0
no exec
line vty 0
password nErdKnoBs
line vty 1
password nErdKnoBs
line vty 2
!
end

```

Figure 2-35 Complete Example Configuration for Router-Eng

X.25 WAN Router Initial Installation Problems

A common problem when bringing new internetworking nodes on-line is that systems on one side of the new node often are unable to communicate with systems on the other side. The problem scenario that follows explores this kind of situation in the context of a private X.25 WAN. In this case, several problems are uncovered during troubleshooting before a final resolution is achieved.

Symptoms

No traffic of any kind can pass through a newly installed router used to interconnect an Ethernet-based network segment to a private X.25 wide area network (WAN). Local area networks (LANs) previously interconnected via the X.25 WAN continue to communicate without disruption of service. However, users trying to make connections cannot get through to resources on the new segment.

Environment Description

Figure 2-36 illustrates a map of an X.25 WAN. The following list summarizes relevant elements of this internetworking environment:

- WAN service is provided to geographically separated networks via a private X.25 packet-switching network.
- Three routers (Router-A, Router-B and Router-C) provide WAN interconnection for hosts and users at three sites (Site-A, Site-B, and Site-C).
- A fourth router (Router-New) has been added to provide WAN interconnection service between a fourth location (Site-New) and the other three sites.
- All four sites are connected to the X.25 network via a fractional T1 service providing 56 Kbps of bandwidth. The routers attach to a CSU/DSU with a RS-449 cable.
- The LANs were all IEEE 802.3 Ethernets.
- The only network layer protocol running in this network is IP; the network uses IGRP to route traffic among IP subnets, with traffic routed over the X.25 links using static address mapping.
- The network applications running over the WAN are limited to file transfer (FTP), mail (SMTP), and virtual terminal connections (Telnet).

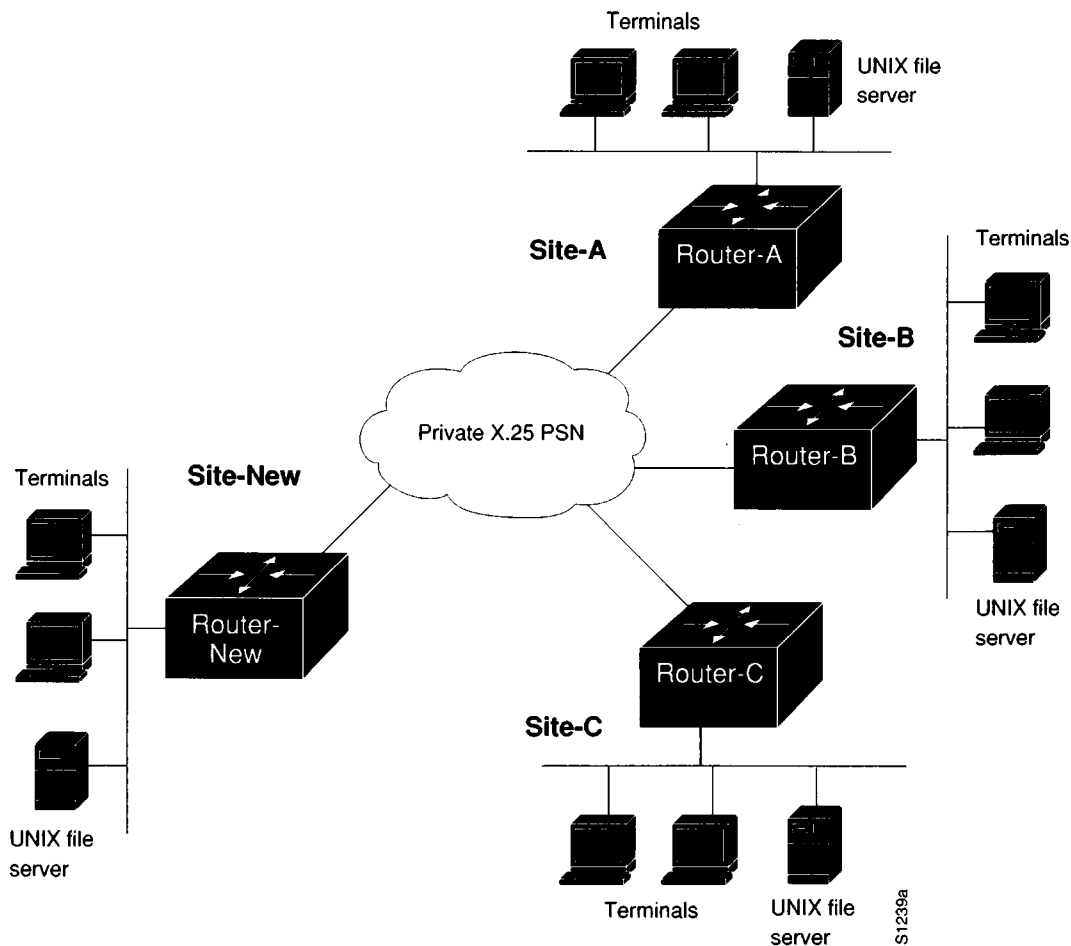


Figure 2-36 X.25 WAN Connectivity Scenario Map

Diagnosing and Isolating Problem Causes

Given this situation, the following problems are the best candidates for interconnection failure:

- Cabling problem to the switch or to the LAN
- Wrong applique (must be DTE for CSU/DSU connectivity)
- Router hardware problem
- Disabled port on the X.25 switch
- Bad T1 digital link
- Mismatched Ethernet version configurations
- Misconfigured hosts
- Software configuration errors

Next, eliminate each potential cause as a problem source and then test the network to determine whether it is operational. The following discussion works through the problem isolation process.

Isolating Serial Hardware and Media Problems

The following procedure illustrates the process of isolating hardware-related problems.

Step 1: The first thing to do is determine the condition of the router. Use the **show version** command. Figure 2-37 illustrates a typical display that the system returns when interfaces are minimally operational and the system can communicate with them. In this case, the interface of interest is associated with an MCI controller.

```
GS Software (GS3-BFX), Version 8.3(1),  
Copyright (c) 1986-1991 by cisco Systems, Inc.  
Compiled Mon 21-Oct-91 22:14 by block  
  
System Bootstrap, Version 4.4(1),  
  
Router-New uptime is 1 day, 22 hours, 19 minutes  
System restarted by power-on  
Running default software  
  
CSC3 (68020) processor with 4096K bytes of memory.  
X.25 software.  
Bridging software.  
1 MCI controller.  
1 cBus controller.  
8 Ethernet/IEEE 802.3 interface.  
1 Token Ring/IEEE 802.5 interface.  
2 Serial network interface.  
1 UltraNet interface.  
1 FDDI network interface.  
Environmental Controller.  
64K bytes of multibus memory.  
64K bytes of non-volatile configuration memory.  
Configuration register is 0x0
```

Figure 2-37 Display Output of Show Version Command

Step 2: In addition to the basic information provided in the **show version** output, use the **show controllers** command to examine the types of appliques on a router and the status of the appliques. Figure 2-38 illustrates an example output of the **show controllers mci** command. In this case, the environment requires a DTE applique to attach the router to a CSU/DSU device. In contrast, a DCE applique typically would be required if the router were connecting directly to a host (DTE interface).

```

MCI 1, controller type 1.1, microcode version 1.8
  128 Kbytes of main memory, 4 Kbytes cache memory
16 system TX buffers, largest buffer size 1520
  Restarts: 0 line down, 0 hung output, 0 controller error
  Interface 0 is Ethernet0, station address 0000.0c00.2be9
    22 total RX buffers, 9 buffer TX queue limit, buffer size 1520
    Transmitter delay is 0 microseconds
  Interface 1 is Serial0, electrical interface is RS-449 DTE
    22 total RX buffers, 9 buffer TX queue limit, buffer size 1520
    Transmitter delay is 0 microseconds
    High speed synchronous serial interface
  Interface 3 is Serial1, electrical interface is RS-449 DTE
    22 total RX buffers, 9 buffer TX queue limit, buffer size 1520
    Transmitter delay is 0 microseconds
    High speed synchronous serial interface

```

Figure 2-38 Example Output of Show Controllers MCI Command

Step 3: Next, determine whether the interface is operational using the **show interfaces serial** command. Figure 2-39 illustrates the output from this command. This particular output indicates that both the serial interface and line protocol are down. These symptoms suggest that there is either a router hardware problem or a cabling problem. In most new installations, this would point to a cabling error. However, you must be sure to check both.

```

Serial 0 is down, line protocol is down
Hardware type is MCI Serial
Internet address is 131.63.125.10, subnet mask is 255.255.255.0
MTU 1006 bytes, BW 64 Kbit, DLY 20000 usec, rely 255/255, load 1/255
Encapsulation X25, loopback not set
LAPB state is SABMSENT, T1 3000, N1 8232, N2 20, K 7
  VS 0, VR 0, RCNT 0, Remote VR 0, Retransmissions 0
  IFRAMES 0/0 RNRs 0/0 REJs 0/0 SABMs 35/1 FRMRs 0/0 DISCs 0/0
X25 address 408026201500, state R1, modulo 8, idle 0, timer 0, nvc 2
Window size: input 7, output 7, Packet size: input 1024, output 1024
Timers: T20 180 T21 200 T22 180 T23 180 TH 0
Channels: Incoming 1-64 Two-way 1-64 Outgoing 1-64
RESTARTs 0/0 CALLs 0+0/0+0/0+0 DIAGs 0/0
Last input never, output never, output hang never
Output queue 0/40, 0 drops; input queue 0/75, 0 drops
Five minute input rate 0 bits/sec, 0 packets/sec
Five minute output rate 0 bits/sec, 0 packets/sec
  0 packets input, 0 bytes, 0 no buffer
  Received 0 broadcasts, 0 runts, 0 giants
  1 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 1 abort
  35 packets output, 152 bytes
  0 output errors, 0 collisions, 39 interface resets, 0 restarts
  1 carrier transitions

```

Figure 2-39 Show Interface Serial Command Indicating Bad Hardware

Step 4: The next step is to check the hardware. Specific tests to determine whether the hardware is operating normally depend on the system type. For instance, you would inspect the applique LEDs on an AGS+, but with an IGS, you would attach a breakout box to the serial port and check the breakout box status LEDs. Refer to Chapter 1 for some general information about interpreting hardware

LEDs and other diagnostics. Refer to your hardware installation and maintenance publication for specifics.

Step 5: In this case, assume that the router hardware is operational, but that the transmit clock (obtained from the CSU/DSU) is not active. The cable is the most likely problem candidate.

To determine whether it is the cable from the modem to the router or from the modem to the switch, configure the CSU/DSU to operate in local loop mode. This terminates use of the line clock (from the T1 service) and forces the CSU/DSU to use the local clock.

Note: A loopback test should be performed using HDLC mode. X.25 does not support loopbacks.

Step 6: Next, inspect the interface status with the **show interfaces serial** command. Assuming that the line protocol *remains* down, a bad cable connection is extremely likely.

To remedy this problem, replace the cable and inspect the interface. Assume that the result is the display shown in Figure 2-40. This display output indicates that the interface is operational and that the cable is working properly. Unfortunately, traffic still is not getting through. However, the display provides a clue about the nature of the remaining problem. This clue is discussed in the next section.

```
Serial 0 is up, line protocol is up
Hardware type is MCI Serial
Internet address is 131.63.125.10, subnet mask is 255.255.255.0
MTU 1006 bytes, BW 64 Kbit, DLY 20000 usec, rely 255/255, load 9/255
Encapsulation X25, loopback not set
LAPB state is CONNECT, TI 3000, NI 8232, N2 20, K 7
  VS 0, VR 3, RCNT 0, Remote VR 0, Retransmissions 0
  IFRAMES 164657/196822 RNRs 0/0 REJts 38/1 SABMs 42/2 FRMRs 0/0 DISCs 0/0
X25 address 000000132600, state R1, modulo 8, idle 1, timer 0, nvc 2
  Window size: input 7, output 7, Packet size: input 1024, output 1024
  Timers: T20 180 T21 200 T22 180 T23 180 TH 0
  Channels: Incoming 1-64 Two-way 1-64 Outgoing 1-64
  RESTARTs 1/1 CALLs 3092-188/5968+0/0+0 DIAGs 0/0
Last input 0:00:00, output 0:00:00, output hang never
Output queue 0/40, 19 drops; input queue 0/75, 0 drops
Five minute input rate 0 bits/sec, 3 packets/sec
Five minute output rate 1000 bits/sec, 4 packets/sec
  315868 packets input, 17218142 bytes, 0 no buffer
  Received 0 broadcasts, 0 runts, 0 giants
  1 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 1 abort
  378780 packets output, 11611102 bytes
  0 output errors, 0 collisions, 41 interface resets, 0 restarts
  1 carrier transitions
```

Figure 2-40 Show Interfaces Output Indicating Link Is Up After Cable Swap

Isolating Interface, LAN, and Local Host Configuration Problems

At this point, hardware problems associated with the serial connection to the X.25 WAN have been eliminated. However, traffic still is unable to get through Router-New. The following procedure steps through the process of isolating problems associated with the LAN interface, the LAN in general, and network hosts.

Step 1: Now determine the status of the router's LAN interface, the LAN media, and the resources on the LAN. Again, the **show interface** command is useful for inspecting the condition of the interface and determining whether it is communicating with devices on the Ethernet. Figure 2-41 illustrates the output resulting from using the **show interfaces ethernet** command. In this case, the interface is alive and properly connected.

```
Ethernet 0 is up, line protocol is up
Hardware is MCI Ethernet, address is 0000.0c00.4dbb (bia 0000.0c00.4dbb)
Internet address is 131.63.152.20, subnet mask is 255.255.255.0
MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec, rely 255/255, load 1/255
Encapsulation ARPA, loopback not set, keepalive set (10 sec)
ARP type: ARPA, ARP Timeout 4:00:00
Last input 0:00:08, output 0:00:08, output hang never
Output queue 0/40, 0 drops; input queue 2/75, 0 drops
Five minute input rate 0 bits/sec, 0 packets/sec
Five minute output rate 0 bits/sec, 0 packets/sec
 19871 packets input, 3030639 bytes, 0 no buffer
   Received 13166 broadcasts, 0 runts, 0 giants
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
 6829 packets output, 799717 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets, 0 restarts
```

Figure 2-41 Show Interface Ethernet Output for Operational Interface

Step 2: Although the output in Figure 2-41 indicates that the interface is operational and that it is seeing traffic on the network, it does not indicate whether the router is able to communicate with specific end nodes on the Ethernet or whether the host configuration allows the host to communicate with the router. To determine whether the host can reach the router, use the **ping** and **clear** commands in coordination with the *Ping* function on the UNIX end system.

First, use the **ping** privileged EXEC command to verify that the router can communicate with each host on the local Ethernet. Figure 2-42 illustrates a successful acknowledgment (Echo Reply) to the Internet Control Message Protocol (ICMP) Echo Request (Ping).

```
Router-New#ping 131.63.152.21
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 131.63.152.21, timeout is 2 seconds:
!!!!
Success rate is 100 percent, round-trip min/avg/max = 1/3/4 ms
```

Figure 2-42 Successful First Ping Communication from Router-New to Target Host

- Step 3:** Step 2 verified that the router is able to communicate with a specific host. However, to verify that the host configuration is correctly specified, ping the router from the host.
- Step 4:** Next, use the **clear arp** privileged EXEC command on the router and **ping** from the router to the host. Figure 2-43 illustrates the successful **ping** transmission and acknowledgment.

```
Router-New#ping 131.63.152.21
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 131.63.152.21, timeout is 2 seconds:
.!!!!
Success rate is 80 percent, round-trip min/avg/max = 1/3/4 ms
```

Figure 2-43 Transmission of Second Ping to Target Host After Clearing ARP Cache

The **ping** is successful, which demonstrates that the host can reply to the router. All LAN-related and host configuration problems are now eliminated. It is time to examine the router's configuration.

Figure 2-44 and Figure 2-45 illustrate the effect of the **ping** exchange on Router-New's ARP cache (*after* the **clear arp** command was executed). Figure 2-44 illustrates that before the **ping** transmission, the ARP cache does not include the target host. After the **ping**, the ARP entry for the host is included in the ARP cache for Router-New (Figure 2-45).

In the second **ping** exchange (refer to Figure 2-43), only 80 percent of the returns are successful. This is the *expected* behavior. Because the end station is not in the original ARP table, the first ping packet is dropped and an ARP request is substituted instead. After the station replies, the subsequent pings work.

```
Router-New#show arp
```

Protocol	Address	Age (min)	Hardware Addr	Type	Interface
Internet	131.63.152.20	0	0000.0c00.f614	ARPA	Ethernet0

Figure 2-44 Example Output of Show Arp Command Before Ping

```
Router-New#show arp
```

Protocol	Address	Age (min)	Hardware Addr	Type	Interface
Internet	131.63.152.21	0	0000.0c00.4dbb	ARPA	Ethernet0
Internet	131.63.152.20	0	0000.0c00.f614	ARPA	Ethernet0

Figure 2-45 Example Output of Show Arp Command After Ping

Isolating Router Software Configuration Problems

Traffic still is not traversing the router. After eliminating all serial hardware problems, LAN problems, and host configuration problems, it appears that a router configuration problem exists. In fact, there may be more than one.

Step 1: The first step in evaluating the router software configuration is to enable X.25 debugging on the router. Use the **debug x25-events** privileged EXEC command. If you intend to keep the output of the **debug** command, you must remember to spool the output to a file. The procedure for setting up such a **debug** output file is described at the beginning of Chapter 10, "Debug Command Reference."

Given the situation, the router is likely to immediately report events. The first to appear are call requests, next are facilities, and finally, cause and diagnostic codes. Figure 2-46 illustrates the **debug** output for this scenario. This information suggests that an invalid virtual circuit is being attempted. Refer to Appendix A, "X.25 Cause and Diagnostics," for more information about specific fields presented in this **debug** output.

X.25 cause and diagnostic output indicates that an invalid virtual circuit is being attempted.

```
Serial0 : X25 I P1 CALL REQUEST (28) 8 lci 1024
From(12): 408026201500 To(12): 000000132600
Facilities: (10)
  Packet size: 1024 1024
  Window size: 7 7
Serial0 : X25 first byte of call user data (1): 0xCC
Serial0 : X25 O P4 CALL CONNECTED (3) 8 lci 1024
Serial0 : X25 I D1 RESET REQUEST (5) 8 lci 1024 cause 3 diag 36
Serial0 : X25 O D1 RESET CONFIRMATION (3) 8 lci 1024

Serial0 : X25 O P2 CALL REQUEST (31) 8 lci 1012
From(12): 000000132600 To(12): 000001241800
Facilities: (10)
  Window size: 7 7
  Packet size: 1024 1024
Serial0 : X25 first byte of call user data (4): 0xCC
Serial0 : X25 I P2 CALL CONNECTED (23) 8 lci 1012
Serial0 : X25 I D1 RESET REQUEST (5) 8 lci 1012 cause 3 diag 36
Serial0 : X25 O D1 RESET CONFIRMATION (3) 8 lci 1012

Serial0 : X25 I P1 CALL REQUEST (28) 8 lci 1023
From(12): 408026201500 To(12): 000000132600
Facilities: (10)
  Packet size: 1024 1024
  Window size: 7 7
Serial0 : X25 first byte of call user data (1): 0xCC
Serial0 : X25 O P4 CALL CONNECTED (3) 8 lci 1023
Serial0 : X25 I D1 RESET REQUEST (5) 8 lci 1023 cause 3 diag 36
Serial0 : X25 O D1 RESET CONFIRMATION (3) 8 lci 1023
```

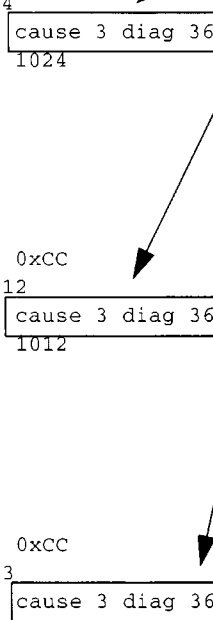


Figure 2-46 Output of Debug X.25 Events Command

Step 2: At this point, you must compare the configuration files of the various routers in the WAN with the configuration file for Router-New.

Contact your X.25 service provider (or network manager) for this information.

There are a number of configurable X.25 parameters that must match for all routers in the WAN. Key parameters include the following:

- X.121 address specification
- Packet size
- Window size
- Virtual circuit channel sequence

In this case, assume that the virtual circuit channel sequencing is not correctly defined in the configuration for Router-New.

The values for Highest Outgoing Channel (HOC), Highest Incoming Channel (HIC), Highest Two-way Channel (HTC), Lowest Outgoing Channel (LOC), Lowest Incoming Channel (LIC), and Lowest Two-way Channel (LTC) are all set to the default in Router-New. However, the WAN requires that these be specifically configured. Because of this mismatch, the router is unable to complete any virtual circuits.

Fixing these errors allows the router to successfully perform ICMP pings, but regular traffic still is not getting through.

Step 3: Again, the answer is buried in the configuration file. In this case, the **x25 map** command does not include the **broadcast** keyword. Because the IP internetwork uses IGRP (a dynamic routing protocol), the **broadcast** keyword is required. Figure 2-47 summarizes the changes made to the configuration file that *finally* allows traffic through the router.

Note: The configuration requirement for the LIC, LOC, and LTC in this case is driven by the specification of a permanent virtual circuit (PVC). The PVC channel specification must be lower than the LIC value. The default LIC value is 1.

```

Current configuration:
!
enable-password iBrPNBLcH
!
!
!
interface Ethernet 0
ip address 131.63.152.20 255.255.255.0
!
interface Serial 0
ip address 131.63.125.10 255.255.255.0
encapsulation X25
x25 win 7
x25 wout 7
x25 ips 1024
x25 ops 1024
x25 address 408026201500
x25 hic 127
x25 htc 127
x25 hoc 127
x25 lic 4
x25 loc 4
x25 ltc 4
x25 nvc 2
x25 facility window size 7 7
x25 facility packet size 1024 1024
x25 map IP 131.63.125.108 40803010123000 broadcast
x25 map IP 131.63.125.12 40802126001000 broadcast
x25 map IP 131.63.125.132 40802121203200 broadcast
x25 map IP 131.63.125.77 40820635287800 broadcast
x25 pcv 1 IP 131.63.126.77
x25 idle 3
!
router igrp 109
network 131.63.0.0
!
!
ip name-server 255.255.255.255
snmp-server community
snmp-server community public RO
hostname Router-New
logging 129.14.87.76
scheduler-interval 1500
!
end

```

Figure 2-47 Complete Configuration Showing Changes Needed to Pass Traffic

Problem Solution Summary

Introducing new internetworking systems into LAN-to-WAN internets is not a trivial matter. The key to resolving multilayered problems is to address each possible problem individually. In this case, multiple causes involved both media problems and software misconfigurations. Connectivity to Site-New was established after making the following changes:

- A bad V.35 cable was replaced with a new cable.
- The router configuration was modified to accommodate the correct HIC, HOC, HTC, LIC, LOC, and LTC X.25 parameters and allow completion of virtual circuits.
- The **x25 map** command was modified to include the **broadcast** option required to handle dynamic routing over X.25.

Internetworking problems are rarely one-dimensional. Problem isolation requires a certain amount of patience and a methodical approach. It is also important to note that subtle protocol variations can wreak havoc in networks if not fully accounted for during the initial configuration. Thus, it is critical to coordinate efforts with all responsible organizations—especially when third parties, such as WAN service vendors, are involved.

Chapter 3

Troubleshooting Apple Connectivity

3

AppleTalk Internetworking Terminology 3-1

- Networks and Internets 3-1
- Phase 1 and Phase 2 Routers 3-2
- Nonextended and Extended Networks 3-2

AppleTalk Internetworking Diagnostic Tips 3-3

- Common AppleTalk Internetworking Problems 3-3
 - Configuration Mismatch 3-4
 - Duplicate Network Numbers 3-5
 - Phase 1/Phase 2 Rule Violations 3-5
 - ZIP Storms 3-6
 - Access List Errors 3-6
 - Unstable Routes (Route Flapping) 3-7
 - Unexpected Back Door 3-7

Preventing AppleTalk Configuration Problems 3-7

- System Startup Precautions 3-10
- Internet Reconfiguration Problem Prevention 3-10

Common AppleTalk Problem Diagnostics 3-10

AppleTalk Connectivity Symptoms 3-12

- Symptom Summary 3-12

Users Cannot See Zones or Services on Remote Networks 3-13

- Possible Causes and Suggested Actions 3-13

Services on a Network Not Visible to Other Networks 3-14

- Possible Causes and Suggested Actions 3-14

Users Cannot Access Services on Remote Networks 3-16

- Possible Causes and Suggested Actions 3-16

Some Zones Missing from Macintosh Chooser 3-18

- Possible Causes and Suggested Actions 3-18

Services Not Always Available; Fade In and Out 3-20

- Possible Causes and Suggested Actions 3-20

Services Visible, but Users Cannot Connect 3-22

- Possible Causes and Suggested Actions 3-22

Zone List Changes Each Time Chooser Is Opened 3-23

Possible Causes and Suggested Actions 3-23

Connections to Services Drop 3-24

Possible Causes and Suggested Actions 3-24

Port Seems Stuck in Restarting or Acquiring Mode 3-25

Possible Causes and Suggested Actions 3-25

Old Zone Names Still Appear in the Chooser 3-26

Possible Causes and Suggested Actions 3-26

Chapter 3

Troubleshooting Apple Connectivity

3

This chapter presents protocol-related troubleshooting information for AppleTalk connectivity problems. The emphasis here is on symptoms and problems associated with AppleTalk network connectivity.

This chapter consists of the following sections:

- Introductory troubleshooting information, including brief definitions of key AppleTalk terms, several diagnostic hints, and summary of the most common AppleTalk internet problems
- An overview of problem prevention techniques
- Summary of AppleTalk symptoms
- Symptom/problem/action modules

The symptom/problem/action modules consist of the following sections:

- Symptom statement—A specific symptom associated with the technology/media/protocol in which this module appears.
- Possible causes and suggested actions—For each symptom, a table containing possible causes for the symptom and suggested actions for resolving each cause.

AppleTalk Internetworking Terminology

The following discussion establishes a framework for discussing AppleTalk internetworking problems. For the purposes of this document, subsequent descriptions adhere to these general rules.

Networks and Internets

It is difficult to distinguish problems that occur on a single cable segment versus those on an entire enterprise network without making an explicit distinction between *networks* and *internets*. For this discussion, an *internet* refers to the entire collection of networks connected via internet routers. *Networks* are individual networks as defined by their associated, unique Appletalk network numbers or cable ranges.

Phase 1 and Phase 2 Routers

There is often confusion over the use of the terms *Phase 1* and *Phase 2* in Appletalk. Cisco refers to *routers* as being Phase 1 or Phase 2 with respect to their ability to support the Appletalk Phase 2 enhancements. Cisco routers dynamically determine whether their neighbors are Phase 2 compliant or not, and operate in Phase 1 compatibility mode if necessary. Most routers currently offered are Phase 2 routers. Older routers that have not been upgraded may be Phase 1 routers.

Note: Some routers can be configured for Phase 1, Phase 2, or *transition* support. Cisco recommends that routers be configured for Phase 2 at the earliest opportunity, subject to limitations in software (such as routers not allowing nonextended Ethernet configurations for Phase 2). Cisco recommends against the use of *transition mode*, which is an interim solution at best. Transition mode implementations can be avoided by using enhancements available in Cisco routers.

Nonextended and Extended Networks

To describe a network or interface, Cisco uses the terms *nonextended* and *extended*. An extended network is one that can contain multiple consecutive network numbers (in other words, a cable range). A nonextended network is one that contains a single network number (such as network 2). Nonextended networks use ARPA (Ethernet Type II) encapsulation on Ethernet. Extended networks use SNAP encapsulation. In addition, FDDI, Token Ring, and most other new media use SNAP encapsulation. An extended network does not require use of multiple network numbers (in other words, 3-3 is a valid extended cable range).

Note: As a further point of clarification, there are no inherent problems in transporting traffic from extended networks across nonextended networks (a common misconception). However, there are certain implementation *rules* that apply to internets featuring both Phase 1 and Phase 2 routers. These rules are discussed later in this chapter.

AppleTalk Internetworking Diagnostic Tips

Internetworks based on the AppleTalk networking protocol suite can encompass complex environments. The fact that they have been designed to make life easier for users does not necessarily make them easier to administer. Before exploring specific symptoms, the following discussions outline some hints and suggestions for AppleTalk internet troubleshooters.

There are two general rules to remember when setting up an AppleTalk internetwork:

1. Every router connected to a specific network must agree on that network's configuration (here, network refers to a single cable segment).
2. Every network number on an internetwork must be unique.

Common AppleTalk Internetworking Problems

This chapter is *primarily* organized around symptoms. In subsequent symptom modules, a list of known possible causes and suggested actions is provided based on a identifiable symptoms. However, many of the most common problems can result in a variety of symptoms. The following discussion covers some the most common problems associated with AppleTalk internets.

The problems summarized here include the following:

- Configuration mismatch
- Duplicate network numbers
- Phase 1/Phase 2 rule violations
- ZIP storms
- Access list errors
- Unstable routes
- Unexpected back doors

These descriptions outline the general nature of each problem and provide some diagnostic notes. Specific actions associated with each problem are detailed in the symptom modules that include these problems as likely causes.

Note: The problems that follow certainly do not represent all known AppleTalk internetworking problems. Indeed, this is only a small subset of potential pitfalls. However, they do represent many of the problems most commonly encountered when creating, upgrading, or modifying AppleTalk internets.

Configuration Mismatch

A configuration mismatch occurs when the following *golden rule* of Appletalk is violated:

All routers on a given cable must agree on the configuration of that cable (meaning that all must have matching network numbers, cable ranges, zone names, and/or zone lists).

To protect against configuration errors in which this rule is violated, many vendors (including Cisco) block activation of any port on which a violation of this rule exists. At interface initialization, if other routers on the network are not in agreement with the way a Cisco router is configured, the Cisco router will not allow AppleTalk to become operational on the interface where a disagreement exists. Cisco routers attempt to restart such an interface every two minutes to avoid outages resulting from transient conditions.

However, if the router is already operational, and another router becomes active whose configuration does not match, the router will continue to operate on that interface until it is restarted. At that point, the interface will fail to become active, and the router will declare a port configuration mismatch in **show appletalk interface**.

Figure 3-1 illustrates a typical example of the **show appletalk interface** display showing the *net.node* address of the device with which the router disagrees.

Indicates port configuration mismatch and shows which neighbor is in conflict.

```
Ethernet 0 is up, line protocol is up
  AppleTalk routing disabled, Port configuration mismatch
  AppleTalk cable range is 4-5
  AppleTalk address is 4.252, Valid
  AppleTalk zone is "Living Dead"
  AppleTalk port configuration conflicts with 4.156
  AppleTalk discarded 8 packets due to input errors
  AppleTalk discarded 2 packets due to output errors
  AppleTalk route cache is disabled, port initializing
```




Figure 3-1 Example Show AppleTalk Interface Display Illustrating Port Mismatch

In Software Release 9.0 and higher, an option is available to display the Name Binding Protocol (NBP) registered name of the conflicting router. This can simplify resolution of a port mismatch problem.

To obtain registered NBP names, enable the **appletalk name-lookup-interval** global configuration command. If enabled, the **show appletalk interfaces** command will display nodes by NBP registration name if available. Refer to your *Router Products Configuration and Reference* publication for more information.

Unstable Routes (Route Flapping)

On very busy internetworks with many routers, it is possible that some routers will fail to send RTMP updates every 10 seconds as they should (due to the excessive load). This results in unnecessary route changes because routes begin to be aged out once two successive RTMP updates have been lost. If severe enough, zones may fade in and out of the Chooser or exhibit other unpredictable behavior. Route instability associated with load problems is known as *flapping*.

Unexpected Back Door

A *back door* is any unexpected path or route through an internetwork. The existence of a back door can result from a number of different events: IP gateways establishing a DDP/IP link unexpectedly; bridges being installed without notice; or even users connecting networks with dial-up connections. Back doors typically cause a change in performance over the internetwork and/or connectivity problems. Performance problems usually occur because all traffic between two sites is going through a lower-bandwidth circuit, or because all traffic is being sent through a single gateway. Connectivity problems can result when routing loops form, or duplicate network numbers are introduced.

Preventing AppleTalk Configuration Problems

Table 3-2 provides a list of suggestions intended to help reduce problems when configuring a router for AppleTalk.

Table 3-2 AppleTalk Problem Prevention Suggestions

Preventive Action	Comments
Upgrade to Phase 2 wherever possible.	This is not a simple task (as it is often described), but is well worth the effort. Pay special attention to upgrading all routers to the same (and most recent default) software version. This will minimize interoperability problems.
When configuring (or making changes to) a router or interface for AppleTalk, enable the command debug apple-events .	<p>This command tracks the progress and status of changes in the internet and alerts you to any errors. You also should run this debug periodically when you suspect network problems. However, in a stable network, this command does not return any information. Remember to disable this debug command with the undebug apple-events command when you have completed diagnostic activities.</p> <p>You may want to consider adding the configuration command appletalk event-logging and establishing a <i>syslog server</i> at your site. This will keep a running log, with timestamps, of significant events on your network.</p>

Preventive Action	Comments
<p>When changing a zone name to an existing network:</p> <p>Step 1: Take all routers for that cable off line for at least 10 minutes. This allows all routers in the internet to age out the network number from their routing tables.</p> <p>Step 2: Configure the new zone list</p> <p>Step 3: Bring the routers back on line.</p>	<p>These actions are recommended because AppleTalk makes no provisions for informing neighbors in the internetwork about a new zone name. Routers only make ZIP queries when a new or previously aged-out network appears on the internet.</p> <p>Adding a new zone to an extended cable configuration will normally result in the router shutting down its interface for Appletalk; its configuration no longer matches that of its neighbors, resulting in a configuration mismatch error.</p>
<p>Use the appletalk timers global configuration command in busy networks with large numbers of internet routers on a single network.</p>	<p>On very busy networks with many LocalTalk-to-EtherTalk routers, the LocalTalk Link Access Protocol (LLAP) routers may not send RTMP updates every 10 seconds as they should. This results in unnecessary route flapping. Suggested value to start: apple timers 10 30 90. The first number should always be 10, and the third number should always be three times the second value. However, setting the second and third numbers to excessively high values can result in slow routing convergence when network topology changes.</p> <p>Timers should be consistently set to the <i>same value</i> throughout the internet, or at a minimum, throughout the internet's <i>backbone</i>.</p>
<p>Minimize the number of different zones in the internet.</p>	<p>Give all of the backbone/WAN connections the same zone name (such as WanZone) or have WAN connections share the zone name of the smaller of the two sites that it connects.</p> <p>In most internetworks, it is not desirable to have the zone names for all backbone or WAN connections appear in the Chooser list. If you make the zone name of all the WAN links the same (WanZone), only that entry appears in the Chooser menu.</p>
<p>Design your network with special attention to the direction in which user/application traffic will flow.</p>	<p>Careful design of the zone mapping can minimize unnecessary NBP traffic. Note that in System 6, if a user opens the Chooser, the Macintosh continually sends NBP BrReq packets. In System 7, Apple has modified this behavior with a logarithmic backoff to minimize the amount of traffic generated.</p> <p>Taking this action can be particularly important in wide area networks (WANs) where traffic traversing WAN links (such as X.25) can be quite expensive.</p>

Preventive Action	Comments
Zones should be named for the convenience of end users and not for diagnostic purposes.	Zones should not be used as cable labels (in other words, do not identify one zone per cable with names like "Bld2 S/W Serial T1"). In general, a mixture of location and departmental naming conventions works best (for example, "Bldg 13 Engineering").
Control the number of zones used.	Many routers have specific limits on the number of routes and zones they can handle. These limits usually result from memory constraints, but are sometimes fixed limits. If you exceed such a limit on a cable connected to one of these devices, zones may come and go unpredictably. Cisco routers do not impose fixed limits.

System Startup Precautions

When bringing a router up on an existing cable where a long zone list is defined, the following actions will help you avoid mistakes and save effort.

1. Bring the interface up in *discovery mode*—**debug apple-events** will let you know when the process is complete. At that time, you will see the following message: *operational*.
2. After discovery is completed, and while in configuration mode, enter the **no appletalk discovery** command for the specific AppleTalk interface being initialized. This action allows the acquired information to be saved and requires that the configuration be validated at port startup. The router exits out of discovery mode for normal operation (it is recommended that discovery mode only be used when initially configuring networks). Thereafter, all routers should be configured for *seed*, or *nondiscovery*, mode.
3. Issue a **write memory** to save the acquired information to nonvolatile RAM.
4. Verify the configuration with **show configuration**.

Internet Reconfiguration Problem Prevention

It is common to create configuration conflicts when changing zone names or cable range numbers. In particular, problems arise when routers exist on the internet about which you are not (administratively) aware.

Remember that many devices can act as routers (for example, Pathworks servers or UNIX workstations running CAP to do print sharing and/or file sharing). In general, if you are changing zone names or cable range numbers in your internetwork, all routers should be shut down or a Cisco router will see a conflict and prevent AppleTalk from initializing on the interface.

Use a network analyzer to listen for router traffic, shut down all routers, wait 10 minutes, and bring up the master seed router.

Common AppleTalk Problem Diagnostics

The following suggestions from router technical support representatives are offered to help speed problem diagnosis and ensure efficient data gathering in the event of failures.

1. The diagnostic command **debug apple-events** is completely silent in a stable network. If it results in any output, unnecessary changes are occurring on the internetwork. You can continuously log the output from this command to a *syslog daemon* on a UNIX host to monitor the internetwork for configuration and status changes.

Duplicate Network Numbers

A very important rule in Appletalk is that network numbers must be unique within an internet, because they comprise the *postal-code* used to route packets. If duplicate network numbers exist, some packets will not be routed to their *intended* destinations and will be lost or misdirected. In addition, duplicate network numbers can cause connectivity and performance problems.

Phase 1/Phase 2 Rule Violations

When Phase 1 and Phase 2 routers are in the same internet, the internet specifications must conform with two mandatory rules:

- There can be no “wide” cable range specifications in the Phase 2 extended portion of the internet. In other words, all cable ranges must span no more than *one* network number. Examples of acceptable cable ranges in this situation are 9-9, 20-20, and 2-2.
- Multiple zones cannot be assigned to cable ranges.

If these internet *compatibility rules* are not followed, connectivity between the nonextended and extended portions of an internetwork will be degraded or even lost. In particular, services located on nonextended networks serviced by Phase 1 routers will not be visible on the other side of the Phase 1 router.

A key difference between Phase 1 and Phase 2 is the way the Name Binding Protocol (NBP) works. This difference can lead to communication problems between Phase 1 and Phase 2 routers. There are four different types of NBP packets in Phase 2 AppleTalk and three in Phase 1. This difference is a point of much confusion in AppleTalk internetworks that support both Phase 1 and Phase 2. Table 3-1 lists the NBP packet types for AppleTalk Phase 1 and Phase 2.

Table 3-1 Comparison of Phase 1 and Phase 2 NBP Packet Types

Phase 1 NBP Packet	Phase 2 NBP Packet
BrRq (Broadcast Request)	BrRq (Broadcast Request)
LkUp (Lookup)	LkUp (Lookup)
	FwdReq (Forward Request)
LkUp-Reply (Lookup Reply)	LkUp-Reply (Lookup Reply)

As shown in Table 3-1, Forward Request (FwdReq) packets do not exist in Phase 1. Only Phase 2 routers will know what to do with them. Phase 1 routers that receive FwdReq packets quietly drop them.

Note: It is important to remember that just because a router is configured for nonextended networks *does not mean* it is a Phase 1 router. A Cisco router running 8.2 software or higher is a Phase 2-compliant router *regardless* of how the interfaces are configured.

ZIP Storms

Routers use the Zone Information Protocol (ZIP) to exchange zone information, and end systems use it to acquire zone lists. Note that there is no mechanism in AppleTalk to force routers to update zone lists. Once a zone has been acquired, routers do not make a ZIP request again unless the network has aged out of the routing table for some reason. As a result, you must use care when adding or removing zone names from an active network.

A *ZIP storm* occurs when a router has propagated a route for which it currently has no corresponding zone name. When the downstream routers also propagate this route, a ZIP storm will ensue.

Question: How do you know when you have a ZIP storm in progress?

Answer: You will see the AppleTalk traffic counters for the ZIP requests increment very rapidly (**show appletalk traffic**). Use the **debug apple-zip** command to identify the network for which the zone is being requested by neighboring routers.



Caution: Throughout this publication, the use of **debug** commands is suggested for obtaining information about network traffic and router status. Use these commands with great care. In general, it is recommended that these commands only be used under the direction of your router technical support representative when troubleshooting specific problems. Enabling debugging can disrupt operation of the router when internets are experiencing high load conditions. In addition, when you have finished using a **debug** command, remember to disable it with the specific **undebug** command or the **undebug all** command.

Question: How can you correct a ZIP storm?

Answer: The best plan is to configure your internetwork so that you *prevent* ZIP storms. Use of Software Release 9.0(1) or later on Cisco routers will help provide a *firewall* against ZIP storms in your internetwork. A ZIP storm will not propagate beyond a Cisco router running Software Release 9.0(1) or later. If a Cisco router receives a routing update from a neighbor, it does not propagate that new route any further until it has received the zone name for it.

However, if you determine that a ZIP storm is occurring, you must search for the router that injected the network number into the internetwork (and that is causing the excessive ZIP traffic). The router commands **show appletalk traffic** and **show appletalk route** provide information that can help you find suspect nodes. Once you have found an offending node, you must stop it from propagating invalid routes. This may require you to upgrade the node's software.

Access List Errors

If properly used, *access lists* can provide a powerful way to control traffic and limit access to resources on an AppleTalk network. However, if not properly implemented, access lists can lead to a number of failures on your internet. Typical problem symptoms associated with incorrectly specified access lists include services for a particular network not being visible to other networks, zones missing from users' Choosers, and services being visible on Choosers but not being accessible.

Unstable Routes (Route Flapping)

On very busy internetworks with many routers, it is possible that some routers will fail to send RTMP updates every 10 seconds as they should (due to the excessive load). This results in unnecessary route changes because routes begin to be aged out once two successive RTMP updates have been lost. If severe enough, zones may fade in and out of the Chooser or exhibit other unpredictable behavior. Route instability associated with load problems is known as *flapping*.

Unexpected Back Door

A *back door* is any unexpected path or route through an internetwork. The existence of a back door can result from a number of different events: IP gateways establishing a DDP/IP link unexpectedly; bridges being installed without notice; or even users connecting networks with dial-up connections. Back doors typically cause a change in performance over the internetwork and/or connectivity problems. Performance problems usually occur because all traffic between two sites is going through a lower-bandwidth circuit, or because all traffic is being sent through a single gateway. Connectivity problems can result when routing loops form, or duplicate network numbers are introduced.

Preventing AppleTalk Configuration Problems

Table 3-2 provides a list of suggestions intended to help reduce problems when configuring a router for AppleTalk.

Table 3-2 AppleTalk Problem Prevention Suggestions

Preventive Action	Comments
Upgrade to Phase 2 wherever possible.	This is not a simple task (as it is often described), but is well worth the effort. Pay special attention to upgrading all routers to the same (and most recent default) software version. This will minimize interoperability problems.
When configuring (or making changes to) a router or interface for AppleTalk, enable the command debug apple-events .	<p>This command tracks the progress and status of changes in the internet and alerts you to any errors. You also should run this debug periodically when you suspect network problems. However, in a stable network, this command does not return any information. Remember to disable this debug command with the undebug apple-events command when you have completed diagnostic activities.</p> <p>You may want to consider adding the configuration command appletalk event-logging and establishing a <i>syslog server</i> at your site. This will keep a running log, with timestamps, of significant events on your network.</p>

Preventive Action	Comments
<p>When changing a zone name to an existing network:</p> <p>Step 1: Take all routers for that cable off line for at least 10 minutes. This allows all routers in the internet to age out the network number from their routing tables.</p> <p>Step 2: Configure the new zone list</p> <p>Step 3: Bring the routers back on line.</p>	<p>These actions are recommended because AppleTalk makes no provisions for informing neighbors in the internetwork about a new zone name. Routers only make ZIP queries when a new or previously aged-out network appears on the internet.</p> <p>Adding a new zone to an extended cable configuration will normally result in the router shutting down its interface for Appletalk; its configuration no longer matches that of its neighbors, resulting in a configuration mismatch error.</p>
<p>Use the appletalk timers global configuration command in busy networks with large numbers of internet routers on a single network.</p>	<p>On very busy networks with many LocalTalk-to-EtherTalk routers, the LocalTalk Link Access Protocol (LLAP) routers may not send RTMP updates every 10 seconds as they should. This results in unnecessary route flapping. Suggested value to start: apple timers 10 30 90. The first number should always be 10, and the third number should always be three times the second value. However, setting the second and third numbers to excessively high values can result in slow routing convergence when network topology changes.</p> <p>Timers should be consistently set to the <i>same value</i> throughout the internet, or at a minimum, throughout the internet's <i>backbone</i>.</p>
<p>Minimize the number of different zones in the internet.</p>	<p>Give all of the backbone/WAN connections the same zone name (such as WanZone) or have WAN connections share the zone name of the smaller of the two sites that it connects.</p> <p>In most internetworks, it is not desirable to have the zone names for all backbone or WAN connections appear in the Chooser list. If you make the zone name of all the WAN links the same (WanZone), only that entry appears in the Chooser menu.</p>
<p>Design your network with special attention to the direction in which user/application traffic will flow.</p>	<p>Careful design of the zone mapping can minimize unnecessary NBP traffic. Note that in System 6, if a user opens the Chooser, the Macintosh continually sends NBP BrReq packets. In System 7, Apple has modified this behavior with a logarithmic backoff to minimize the amount of traffic generated.</p> <p>Taking this action can be particularly important in wide area networks (WANs) where traffic traversing WAN links (such as X.25) can be quite expensive.</p>

Preventive Action	Comments
Zones should be named for the convenience of end users and not for diagnostic purposes.	Zones should not be used as cable labels (in other words, do not identify one zone per cable with names like “Bld2 S/W Serial T1”). In general, a mixture of location and departmental naming conventions works best (for example, “Bldg 13 Engineering”).
Control the number of zones used.	Many routers have specific limits on the number of routes and zones they can handle. These limits usually result from memory constraints, but are sometimes fixed limits. If you exceed such a limit on a cable connected to one of these devices, zones may come and go unpredictably. Cisco routers do not impose fixed limits.

System Startup Precautions

When bringing a router up on an existing cable where a long zone list is defined, the following actions will help you avoid mistakes and save effort.

1. Bring the interface up in *discovery mode*—**debug apple-events** will let you know when the process is complete. At that time, you will see the following message: `operational`.
2. After discovery is completed, and while in configuration mode, enter the **no appletalk discovery** command for the specific AppleTalk interface being initialized. This action allows the acquired information to be saved and requires that the configuration be validated at port startup. The router exits out of discovery mode for normal operation (it is recommended that discovery mode only be used when initially configuring networks). Thereafter, all routers should be configured for *seed*, or *nondiscovery*, mode.
3. Issue a **write memory** to save the acquired information to nonvolatile RAM.
4. Verify the configuration with **show configuration**.

Internet Reconfiguration Problem Prevention

It is common to create configuration conflicts when changing zone names or cable range numbers. In particular, problems arise when routers exist on the internet about which you are not (administratively) aware.

Remember that many devices can act as routers (for example, Pathworks servers or UNIX workstations running CAP to do print sharing and/or file sharing). In general, if you are changing zone names or cable range numbers in your internetwork, all routers should be shut down or a Cisco router will see a conflict and prevent AppleTalk from initializing on the interface.

Use a network analyzer to listen for router traffic, shut down all routers, wait 10 minutes, and bring up the master seed router.

Common AppleTalk Problem Diagnostics

The following suggestions from router technical support representatives are offered to help speed problem diagnosis and ensure efficient data gathering in the event of failures.

1. The diagnostic command **debug apple-events** is completely silent in a stable network. If it results in any output, unnecessary changes are occurring on the internetwork. You can continuously log the output from this command to a *syslog daemon* on a UNIX host to monitor the internetwork for configuration and status changes.

AppleTalk Connectivity Symptoms

The symptom modules that follow pertain to AppleTalk internetwork problems. Unless otherwise indicated, each module is presented as a set of general problems. Where there are special considerations associated with a situation, notes are included.

Symptom Summary

AppleTalk connectivity symptoms discussed in this section include the following:

- Users cannot see zones or services on remote networks
- Services in a network are not visible outside that network
- Users cannot access services on remote networks
- Zones are missing from Choosers of Macs
- Services not always available; fade in and out
- Services visible, but users cannot connect
- Zone list changes each time Chooser is opened
- Connection to services drop
- Port seems stuck in acquiring mode
- Old zone names still appear in Chooser



Caution: Throughout this publication, the use of **debug** commands is suggested for obtaining information about network traffic and router status. Use these commands with great care. In general, it is recommended that these commands only be used under the direction of your router technical support representative when troubleshooting specific problems. Enabling debugging can disrupt operation of the router when internets are experiencing high load conditions. In addition, when you have finished using a **debug** command, remember to disable it with the specific **undebug** command or the **undebug all** command.

2. To identify problem nodes, you can run **ping** tests with a one-line command. For example, **ping 2.24** will ping AppleTalk node 2.24. Use this command to verify that the node is reachable from the router. The **ping** command also supports a number of AppleTalk variable parameters.

The NBP option of the AppleTalk **ping** function provides additional troubleshooting capabilities. In particular, use the NBP option when you find that AppleTalk zones are listed in the Chooser, but services are not available. In addition the NBP features (implemented as of Software Release 9.0) display nodes by NBP registration name, if enabled using the **appletalk name-lookup-interval** command. Refer to your *Router Products Configuration and Reference* publication for more information.

Users Cannot See Zones or Services on Remote Networks

Symptom: Although users are able to access services on their own network, offnet zones and services expected to be available from their Chooser lists are not accessible.

Possible Causes and Suggested Actions

Table 3-3 outlines possible causes of blocked access to offnet zones and network resources.

Table 3-3 Causes and Actions for Blocked Access to Offnet Resources

Possible Cause	Suggested Actions
Configuration mismatch	<p>Step 1: From the router, perform show appletalk interface.</p> <p>Step 2: Look for “port configuration mismatch” message; indicates that the configuration disagrees with listed neighbor.</p> <p>Step 3: If no error is displayed, execute the clear interface command for the interface in question. If it becomes operational after clearing, a configuration mismatch does not exist. If it declares “port configuration mismatch,” continue with the steps that follow.</p> <p>Step 4: Verify that configuration for each router agrees regarding network number or cable range and the zone or zone list. In some cases, the configuration shown is not the configuration being used, so if problems persist, set the problem router to get its <i>seed</i> information from the network (put the router in discovery mode by specifying appletalk address 0.0).</p> <p>Step 5: If they do not agree, modify configurations of the other routers as necessary.</p> <p>Step 6: If the problem persists, try to determine which router is at fault. The show appletalk interface command will generate the network and node address of the conflicting router. If appletalk name-lookup-interval is enabled, the NBP registration name will be displayed. If you are unable to identify the misconfigured router using the node address, determine the hardware address of the conflicting router with show appletalk arp. This also allows you to determine the vendor code (vendor codes are available in RFC 1340).</p> <p>Step 7: As an alternative, all routers but one can be configured for nonseed or discovery mode, then routers can be restarted.</p>

Services on a Network Not Visible to Other Networks

Symptom: Users find that the AppleTalk services for a particular network do not appear in their Choosers.

Possible Causes and Suggested Actions

Table 3-4 outlines possible causes of missing network services.

Table 3-4 Causes and Actions for Missing Services

Possible Cause	Suggested Actions
Configuration mismatch	<p>Step 1: From the router, perform show appletalk interface.</p> <p>Step 2: Look for “port configuration mismatch” message; indicates that the configuration disagrees with listed neighbor.</p> <p>Step 3: If no error is displayed, execute the command clear interface for the interface in question. If it becomes operational after clearing, a configuration mismatch does not exist. If it declares “port configuration mismatch,” continue with the steps that follow.</p> <p>Step 4: Verify that configuration for each router agrees regarding network number or cable range and the zone or zone list. In some cases, the configuration shown is not the configuration being used, so if problems persist, set the problem router to get its <i>seed</i> information from the network (put the router in discovery mode by specifying appletalk address 0.0).</p> <p>Step 5: If they do not agree, modify configurations of the other routers as necessary.</p> <p>Step 6: If the problem persists, try to determine which router is at fault. The show appletalk interface command will generate the network and node address of the conflicting router. If appletalk name-lookup-interval is enabled, the NBP registration name will be displayed. If you are unable to identify the misconfigured router using the node address, determine the hardware address of the conflicting router with show appletalk arp. This also allows you to determine the vendor code (vendor codes are available in RFC 1340).</p> <p>Step 7: As an alternative, all routers but one can be configured for nonseed or discovery mode, then the routers can be restarted.</p>

Possible Cause	Suggested Actions
Duplicate network numbers	<p>Step 1: The network where these symptoms are noticeable is likely to contain a duplicate network number.</p> <p>Either change the network number of the afflicted network or remove Appletalk from the suspect/problem interface. In either case, the interface's original network number should disappear from the internet within a few minutes. If it persists, you probably have found the duplicate network.</p> <p>Step 2: If you changed the network number on the interface, no further action is required. If not, change it now (make sure it is unique). Remember to reenter the zone name and any other interface configurations for Appletalk on that interface.</p>
Phase 1/Phase 2 rule violations	<p>Step 1: Use show appletalk global to determine whether or not the internet is in <i>compatibility mode</i>.</p> <p>Step 2: Use show appletalk neighbor to determine which specific neighbor is in compatibility mode (if appletalk name-lookup-interval is enabled, shown by NBP name).</p> <p>Step 3: Select one of three solutions:</p> <ul style="list-style-type: none"> ■ Ensure that all routers are in compliance with the two Phase 1/Phase 2 rules. ■ Upgrade AppleTalk Phase 1 routers to AppleTalk Phase 2 compliance and reconfigure the internet. ■ Use appletalk proxy-nbp feature. <p>To use appletalk proxy-nbp, you must create at least one <i>virtual</i> network on the router that has the same zone name as the network where the unreachable services exist. This forces the router to use Phase 1-type NBP lookups (in addition to Phase 2-style FwdRequests) when sending NBP requests through the network. Since the Lookup is defined for Phase 1 routers, the Phase 1 router will properly route the request on to the service and a reply should be received. Refer to the <i>Router Products Configuration and Reference</i> publication for a more information.</p>
Access list errors	<p>Step 1: Carefully disable access lists on suspect routers and see whether connectivity returns.</p> <p>Step 2: If connectivity returns, access list error is likely suspect. Check access lists and associated configuration commands for errors.</p> <p>Step 3: Modify any access lists as necessary.</p> <p>Step 4: If connection problems persist, consult with your router technical support representative for more assistance.</p>

Users Cannot Access Services on Remote Networks

Symptom: Users on a particular network find that they cannot access services off their network, and network administrators discover that the router interface connected to their network will not initialize AppleTalk operation.

Possible Causes and Suggested Actions

Table 3-5 outlines possible causes of an AppleTalk interface failing to initialize.

Table 3-5 Causes and Actions for Interface Failing to Start AppleTalk

Possible Cause	Suggested Actions
Configuration mismatch	<p>Step 1: From the router, perform show appletalk interface.</p> <p>Step 2: Look for “port configuration mismatch” message; indicates that the configuration disagrees with listed neighbor.</p> <p>Step 3: If no error is displayed, execute the command clear interface for the interface in question. If it becomes operational after clearing, a configuration mismatch does not exist. If it declares “port configuration mismatch,” continue with the steps that follow.</p> <p>Step 4: Verify that configuration for each router agrees regarding network number or cable range and the zone or zone list. In some cases, the configuration shown is not the configuration being used, so if problems persist, set the problem router to get its <i>seed</i> information from the network (put the router in discovery mode by specifying appletalk address 0.0).</p> <p>Step 5: If they do not agree, modify configurations of the other routers as necessary.</p> <p>Step 6: If the problem persists, try to determine which router is at fault. The show appletalk interfaces command will generate the network and node address of the conflicting router. If apple name-lookup-interval is enabled, the NBP registration name will be displayed. If you are unable to identify the misconfigured router using the node address, look up the hardware address of the conflicting router with show appletalk arp. This display also provides the node’s vendor code (vendor codes are available in RFC 1340).</p> <p>Step 7: As an alternative, all routers but one can be configured for nonseed or discovery mode, then the routers can be restarted.</p>

Possible Cause	Suggested Actions
Phase 1/Phase 2 rule violations	<p>Step 1: Use show appletalk global to determine whether or not the internet is in <i>compatibility mode</i>.</p> <p>Step 2: Use show appletalk neighbor to determine which specific neighbor is in compatibility mode (if appletalk name-lookup-interval is enabled, shown by NBP name).</p> <p>Step 3: Select one of three solutions:</p> <ul style="list-style-type: none"> ■ Ensure that all routers are in compliance with the two Phase 1/Phase 2 rules. ■ Upgrade AppleTalk Phase 1 routers to AppleTalk Phase 2 compliance and reconfigure the internet. ■ Use appletalk proxy-nbp feature. <p>To use appletalk proxy-nbp, you must create at least one <i>virtual</i> network on the router that has the same zone name as the network where the unreachable services exist. This forces the router to use Phase 1-type NBP lookups (in addition to Phase 2-style FwdRequests) when sending NBP requests through the network. Since the Lookup is defined for Phase 1 routers, the Phase 1 router will properly route the request on to the service and a reply should be received. Refer to the <i>Router Products Configuration and Reference</i> publication for a more information.</p>

Some Zones Missing from Macintosh Chooser

Symptom: Mac users on different networks report that zones associated with a particular network do not appear in their Chooser listings.

Possible Causes and Suggested Actions

Table 3-6 outlines possible causes of zones not appearing in the Chooser on networks separated by a router.

Table 3-6 Causes and Actions for Zones Not Appearing

Possible Cause	Suggested Actions
Configuration mismatch	<p>Step 1: From the router, perform show appletalk interface.</p> <p>Step 2: Look for “port configuration mismatch” message; indicates that the configuration disagrees with listed neighbor.</p> <p>Step 3: If no error is displayed, execute the command clear interface for the interface in question. If it becomes operational after clearing, a configuration mismatch does not exist. If it declares “port configuration mismatch,” continue with the steps that follow.</p> <p>Step 4: Verify that configuration for each router agrees regarding network number or cable range and the zone or zone list. In some cases, the configuration shown is not the configuration being used, so if problems persist, set the problem router to get its <i>seed</i> information from the network (put the router in discovery mode by specifying appletalk address 0.0).</p> <p>Step 5: If they do not agree, modify configurations of the other routers as necessary.</p> <p>Step 6: If the problem persists, try to determine which router is at fault. The show appletalk interface command will generate the network and node address of the conflicting router. If apple name-lookup-interval is enabled, the NBP registration name will be displayed. If you are unable to identify the misconfigured router using the node address, look up the hardware address of the conflicting router with show appletalk arp. This display also provides the node’s vendor code (vendor codes are available in RFC 1340).</p> <p>Step 7: Alternately, all routers but one can be configured for nonseed or discovery mode, then the routers can be restarted.</p>

Possible Cause	Suggested Actions
ZIP storm	<p>Step 1: Use show appletalk traffic to look for the number of ZIP Requests displayed; repeat after about 30 seconds.</p> <p>Step 2: Compare resulting display output. If number is greater than 10 and increasing, a ZIP storm probably is occurring.</p> <p>Step 3: Use show appletalk route to see whether a network shows up in the table, even though the zone indicates no zone set in the display.</p> <p>Step 4: If you find a network with this zone specification, the node in that network is probably not responding to ZIP requests, resulting in the ZIP storm.</p> <p>Step 5: Determine why the node is not responding to ZIP requests.</p> <p>Step 6: ZIP storms may result from a defect in the problem node's software. Contact the vendor to determine whether there is a known problem.</p>
Access list errors	<p>Step 1: Carefully disable access lists on suspect routers and see whether connectivity returns.</p> <p>Step 2: If connectivity returns, access list error is likely suspect. Check access lists and associated configuration commands for errors.</p> <p>Step 3: Modify any access lists as necessary.</p> <p>Step 4: If connection problems persist, consult with your router technical support representative for more assistance.</p>
Unstable routes	<p>Step 1: Check traffic load with show interface. For interfaces with more than 50 percent load, you may need to further segment network to limit traffic.</p> <p>Step 2: Issue the command debug apple-events to determine whether routes are being aged incorrectly.</p> <p>Step 3: Use the appletalk timers command to correct the problem. Suggested parameter values for the command are 10, 30, and 90 to start, but do not exceed 10, 40, and 120. The first number must always be 10, and the third value should be three times the second.</p> <p>Timers should be consistently set to the <i>same value</i> throughout the internet, or at a minimum, throughout the internet's <i>backbone</i>.</p> <p>This type of problem often can be alleviated by simply segmenting the network to limit the number of routers on a segment.</p>

Services Not Always Available; Fade In and Out

Symptom: Mac users report that services are intermittently unavailable. Services come and go without warning.

Possible Causes and Suggested Actions

Table 3-7 outlines possible causes of intermittent loss of AppleTalk services.

Table 3-7 Causes and Actions for Intermittent AppleTalk Service Loss

Possible Cause	Suggested Actions
Duplicate network numbers	<p>Step 1: Any network where these symptoms are noticeable is likely to be one where a duplicate network number exists.</p> <p>Either change the network number of the afflicted network or remove Appletalk from the suspect/problem interface. In either case, the interface's original network number should disappear from the internet within a few minutes. If it persists, you have likely found the duplicate network.</p> <p>Step 2: If you changed your network number on the interface, no further action is required. If not, change it now (make sure it is unique). Remember to reenter the zone name and any other interface configurations for Appletalk on that interface.</p>
ZIP storm	<p>Step 1: Use show appletalk traffic to look for the number of ZIP Requests displayed; repeat after about 30 seconds.</p> <p>Step 2: Compare resulting display output. If number is greater than 10 and increasing, a ZIP storm is probably occurring.</p> <p>Step 3: Use show appletalk route to see whether a network shows up in the table, even though the zone indicates no zone set in the display.</p> <p>Step 4: If you find a network with this zone specification, the node in that network is probably not responding to ZIP requests, resulting in the ZIP storm.</p> <p>Step 5: Determine why the node is not responding to ZIP requests.</p> <p>Step 6: ZIP storms may result from a defect in the problem node's software. Contact the vendor to determine whether there is a known problem.</p>

Possible Cause	Suggested Actions
Unstable routes	<p>Step 1: Check traffic load with show interface. For interfaces with more than 50 percent load, you may need to further segment network to limit traffic.</p> <p>Step 2: Issue the command debug apple-events to determine whether routes are being aged incorrectly.</p> <p>Step 3: Use the appletalk timers command to correct the problem. Suggested parameter values for the command are 10, 30, and 90 to start, but do not exceed 10, 40, and 120. The first number must always be 10, and the third value should be three times the second.</p> <p>Timers should be consistently set to the <i>same value</i> throughout the internet, or at a minimum, throughout the internet's <i>backbone</i>.</p> <p>This type of problem often can be alleviated by simply segmenting the network to limit the number of routers on a segment.</p>
Overloaded network, where routes are being aged out	<p>Step 1: Check traffic load with show interface.</p> <p>Step 2: For interfaces with more than 50 percent load, you may need to further segment network to limit traffic.</p> <p>Step 3: Issue the command debug apple-events to determine whether routes are being aged incorrectly. Then use the appletalk timers command to correct the problem.</p> <p>Timers should be consistently set to the <i>same value</i> throughout the internet, or at a minimum, throughout the internet's <i>backbone</i>.</p>

Services Visible, but Users Cannot Connect

Symptom: Users report that Apple services appear in their Chooser lists, but they are unable to access the services.

Possible Causes and Suggested Actions

Table 3-8 outlines possible causes of services appearing in Choosers but not being available.

Table 3-8 Causes and Actions for Blocked Service Access

Possible Cause	Suggested Actions
Duplicate network numbers	<p>Step 1: Any network where these symptoms are noticeable is likely to be one where a duplicate network number exists.</p> <p>Either change the network number of the afflicted network or remove Appletalk from the suspect/problem interface. In either case, the interface's original network number should disappear from the internet within a few minutes. If it persists, you have likely found the duplicate network.</p> <p>Step 2: If you changed your network number on the interface, no further action is required. If not, change it now (make sure it is unique). Remember to reenter the zone name and any other interface configurations for Appletalk on that interface.</p>
ZIP storm	<p>Step 1: Use show appletalk traffic to look for the number of ZIP Requests displayed; repeat after about 30 seconds.</p> <p>Step 2: Compare resulting display output. If number is greater than 10 and increasing, a ZIP storm is probably occurring.</p> <p>Step 3: Use show appletalk route to see whether a network shows up in the table, even though the zone indicates no zone set in the display.</p> <p>Step 4: If you find a network with this zone specification, the node in that network is probably not responding to ZIP requests, resulting in the ZIP storm.</p> <p>Step 5: Determine why the node is not responding to ZIP requests.</p> <p>Step 6: ZIP storms may result from a defect in the problem node's software. Contact the vendor to determine whether there is a known problem.</p>
Access list errors	<p>Step 1: Carefully disable access lists on suspect routers and see whether connectivity returns.</p> <p>Step 2: If connectivity returns, access list error is likely suspect. Check access lists and associated configuration commands for errors.</p> <p>Step 3: Modify any access lists as necessary.</p> <p>Step 4: If connection problems persist, consult with your router technical support representative for more assistance.</p>

Zone List Changes Each Time Chooser Is Opened

Symptom: Puzzled users report that whenever their Chooser windows are opened, the zone list appears to change.

Possible Causes and Suggested Actions

Table 3-9 outlines possible causes of zones changing whenever the Chooser window is opened.

Table 3-9 Causes and Actions for Zone List Constantly Changing

Possible Cause	Suggested Actions
Unstable routes	<p>Step 1: Check traffic load with show interface. For interfaces with more than 50 percent load, you may need to further segment network to limit traffic.</p> <p>Step 2: Issue the command debug apple-events to determine whether routes are being aged incorrectly.</p> <p>Step 3: Use the appletalk timers command to correct the problem. Suggested parameter values for the command are 10, 30, and 90 to start, but do not exceed 10, 40, and 120. The first number must always be 10, and the third value should be three times the second.</p> <p>Timers should be consistently set to the <i>same value</i> throughout the internet, or at a minimum, throughout the internet's <i>backbone</i>.</p> <p>This type of problem often can be alleviated by simply segmenting the network to limit the number of routers on a segment.</p>
Routers on network have different zone lists	<p>Step 1: Verify that the configuration of zone lists for all routers are in agreement.</p> <p>Step 2: If they do not agree, reconfigure the routers so that all have matching zone lists for relevant networks.</p>

Connections to Services Drop

Symptom: Users complain that their sessions with Apple services suddenly drop for no apparent reason.

Possible Causes and Suggested Actions

Table 3-10 outlines possible causes of unexpected loss of Apple network services.

Table 3-10 Causes and Actions for Services Being Dropped

Possible Cause	Suggested Actions
Unstable routes	<p>Step 1: Check traffic load with show interface. For interfaces with more than 50 percent load, you may need to further segment network to limit traffic.</p> <p>Step 2: Issue the command debug apple-events to determine whether routes are being aged incorrectly.</p> <p>Step 3: Use the appletalk timers command to correct the problem. Suggested parameter values for the command are 10, 30, and 90 to start, but do not exceed 10, 40, and 120. The first number must always be 10, and the third value should be three times the second.</p> <p>Timers should be consistently set to the <i>same value</i> throughout the internet, or at a minimum, throughout the internet's <i>backbone</i>.</p> <p>This type of problem often can be alleviated by simply segmenting the network to limit the number of routers on a segment.</p>

Port Seems Stuck in Restarting or Acquiring Mode

Symptom: Router is unable to discover routes or poll neighbors on attached cable.

Possible Causes and Suggested Actions

Table 3-11 outlines possible causes for a stuck port problem in a router.

Table 3-11 Causes and Actions for Stuck Port Problem

Possible Cause	Suggested Actions
Inadvertently crossed (serial) circuits with multiple lines between two routers	Step 1: Check physical attachment of serial lines to ensure that they are correctly wired.
	Step 2: If needed, rewire and check show interface and show appletalk interface output to confirm that traffic is getting through and that the protocol is operating correctly.
	Step 3: If router is still unable to find routes, consult your router technical support representative for more assistance.
Router is in discovery mode and no seed router exists on network	Step 1: If there is no seed router on the network, put the router in nondiscovery mode and restart the interface (exit configuration mode). Remember to assign a zone and network number or cable range (assign network number or cable range using the appletalk address or appletalk cable-range interface subcommands).
	Step 2: Use the no appletalk discovery interface command to allow the specific interface(s) to be seed ports.

Old Zone Names Still Appear in the Chooser

Symptom: Users report that they are seeing zones that have been deleted from the network.

Note: Appletalk does not incorporate any provisions to update routing tables when zone names are changed. For example, if the zone name for network number 200 is *Twilight Zone*, but you decide to change the zone to *No Parking Zone*, the zone name on the interface can be changed and the new zone name takes effect locally. However, unless you keep network 200 off the internet long enough for it to be aged out of the internet's routing tables completely, some routers will continue to use the old zone name.

Possible Causes and Suggested Actions

Table 3-12 outlines possible causes of unnumbered zones in AppleTalk interconnection environments.

Table 3-12 Causes and Actions for Zones with Missing Network Numbers.

Possible Cause	Suggested Actions
Configuration mismatch	<p>Step 1: From the router, perform show appletalk interface.</p> <p>Step 2: Look for "port configuration mismatch" message; indicates that configuration disagrees with indicated neighbor.</p> <p>Step 3: If no error is displayed, execute the command clear interface for the interface in question. If it becomes operational after clearing, a configuration mismatch does not exist. If it declares "port configuration mismatch," continue with the steps that follow.</p> <p>Step 4: Verify that configuration for each router agrees regarding network number or cable range and the zone or zone list. In some cases, the configuration shown is not the configuration being used, so if problems persist, set the problem router to get its <i>seed</i> information from the network (put the router in discovery mode by specifying appletalk address 0.0).</p> <p>Step 5: If they do not agree, modify configurations of the other routers as necessary.</p> <p>Step 6: If the problem persists, try to determine which router is at fault. The show appletalk interfaces command will generate the network and node address of the conflicting router. If apple name-lookup-interval is enabled, the NBP registration name will be displayed.</p>

Possible Cause	Suggested Actions	
		If you are unable to determine which router is misconfigured by the node address, looking up the hardware address of the conflicting router with show appletalk arp allows you to determine the vendor to help narrow the search (vendor codes are available in RFC 1340).
	Step 7:	As an alternative, all routers but one can be configured for nonseed or discovery mode, then the routers can be restarted.
<i>Ghost zone</i> created when a zone name is changed without changing the associated network number	Step 1:	Check the network numbers for each AppleTalk interface in the router configuration.
	Step 2:	Ensure that any network numbers associated with an old zone name are removed.
	Step 3:	Check show appletalk zones display output to be sure that the <i>ghost zone</i> no longer appears in the list.

Note: Since AppleTalk has no provision for flushing zones that are not valid, always change the underlying network number when changing the zone name for a cable.

Chapter 4

Troubleshooting IBM Connectivity

4

Diagnosing IBM Network and Token Ring Problems 4-2

IBM Network and Token Ring Connectivity Symptoms 4-3

Routing Does Not Function in SRB Environment 4-4

Possible Causes and Suggested Actions 4-4

Routing in SRB Network Fails Unexpectedly 4-5

Possible Causes and Suggested Actions 4-5

No Communication over SRB 4-6

Possible Causes and Suggested Actions 4-6

Blocked Communication over Remote SRB 4-8

Possible Causes and Suggested Actions 4-8

Intermittent Communication Failures over Remote SRB 4-9

Possible Causes and Suggested Actions 4-9

Users Cannot Communicate over Cisco Translational Bridge 4-10

Possible Causes and Suggested Actions 4-10

Traffic Cannot Get Through Router Implementing SRT 4-12

Possible Causes and Suggested Actions 4-12

Users Cannot Make Connections over Router Configured for SDLLC 4-13

Possible Causes and Suggested Actions 4-13

IBM RS-232 Signaling Requirements Summary 4-14

Preventive Actions in SDLLC Environments 4-15

Virtual Token Ring Addresses and SDLLC Implementations 4-15

Intermittent Connectivity over Router Configured for SDLC 4-16

Possible Causes and Suggested Actions 4-16

Router Is Unable to Connect to Token Ring 4-17

Possible Causes and Suggested Actions 4-17

Router Is Not Communicating with IBM SDLC Devices over RS-232 4-19

Possible Causes and Suggested Actions 4-19

SDLC Sessions Fail over Router Running STUN 4-20

Possible Causes and Suggested Actions 4-20

NetBIOS Devices Cannot Communicate over Remote SRB 4-22

Possible Causes and Suggested Actions 4-22

Router Cannot Be Linked from LAN Network Manager 4-23

Possible Causes and Suggested Actions 4-23

Chapter 4

Troubleshooting IBM Connectivity

4

This chapter focuses on a series of common symptoms associated with routing and bridging in IBM-based networks, possible causes of those symptoms, and general suggestions for identifying, isolating, and resolving those causes. The emphasis here is on symptoms and problems associated with IBM network connectivity.

This chapter consists of the following sections:

- IBM connectivity symptom list
- Symptom/cause/action modules

The symptom/cause/action modules consist of the following sections:

- Symptom statement—A specific symptom associated with the technology/media/protocol in which this module appears
- Possible causes and suggested actions— A table for each symptom containing possible causes for the symptom and suggested actions for resolving each cause.

Note: This chapter focuses on IBM-related and Token Ring problems. General diagnostic tools and techniques used for isolating serial line problems are discussed in Chapter 7, “Troubleshooting WAN Connectivity.”

Diagnosing IBM Network and Token Ring Problems

When troubleshooting connectivity problems in IBM internets, there are a variety of tools and techniques that you can apply to isolate the causes of interruptions.

The following tools are universally applicable when gathering information to troubleshoot IBM internetworking links:

- Output of relevant **show** commands. Some of these include **show rif**, **show arp**, **show source-bridge**, **show controller token**, **show interface token**, **show version**, **show tcp**, **show flash** (useful only if the Flash Memory card is installed in your system), **show lnm bridge**, **show lnm config**, **show lnm interface**, and **show netbios-cache**.
- Output of relevant **debug** commands. Some useful **debug** commands include **debug rif**, **debug source-bridge**, **debug source-event**, **debug lnm-events**, **debug lnm-llc**, **debug lnm-mac**, and **debug netbios-name-cache**.



Caution: Throughout this and other chapters, the use of **debug** commands is suggested for obtaining information about network traffic and router status. Use these commands with great care. In general, it is recommended that these commands only be used under the direction of your router technical support representative when troubleshooting specific problems. Enabling debugging can disrupt operation of the router when internets are experiencing high load conditions. Avoid using **debug** commands in production networks. When you finish using a **debug** command, remember to disable it with its specific **undebug** command or with the **undebug all** command.

- Output of **write terminal** or **show configuration** screens to evaluate operational configuration.
- Output of network analyzer (attached to end-station token ring) trace. In the scenarios that appear in this manual, output is from a Network General *Sniffer*, but this suggests no endorsement of a particular product. This output is for illustrative purposes only.

IBM Network and Token Ring Connectivity Symptoms

The symptom modules that follow pertain to IBM internetworking problems. Unless otherwise indicated, each module is presented as a set of general problems. Where there are special considerations associated with a specific network type, notes are included.

IBM networking connectivity symptoms discussed in this section include the following:

- Routing does not function in SRB environment
- Routing in SRB network fails unexpectedly
- No communication over router configured as an SRB
- Blocked communication over router configured for remote SRB
- Intermittent communication failures over router configured for remote SRB
- Users cannot communicate over router when attempting translational bridging
- Traffic not transiting through router implementing SRT
- Users cannot establish connections over router running SDLLC
- Intermittent connectivity over router running SDLC
- Router is unable to connect to Token Ring
- Router is not communicating with IBM SDLC devices over RS-232
- SDLC sessions fail over router running STUN
- NetBIOS devices cannot communicate over remote SRB
- Router cannot be linked from LAN Network Manager

Routing Does Not Function in SRB Environment

Symptom: In some cases, you may need to route certain protocols (for example, Novell IPX) through internetworks that are dominated by source route bridge (SRB) links. It is not unusual to find that the SRBs are bridging traffic as they should, but the routed protocol(s) is (are) not getting through a router.

Possible Causes and Suggested Actions

Table 4-1 outlines possible problems that block routing of traffic through routers in SRB internetworks.

Table 4-1 Causes and Actions for Blocked Routing in SRB Environments

Possible Cause	Suggested Actions
Misconfigured router while routing a protocol and attempting to communicate with host on another ring across an SRB	<p>Step 1: Check configuration for inclusion or absence of multiring protocol-name command.</p> <p>Step 2: Add this command to the router configuration if it is missing.</p> <p>Step 3: Try the steps outlined in the next symptom, "Routing in SRB Network Fails Unexpectedly."</p> <p>Step 4: Contact your router technical support representative if you are still unable to get traffic intended to be routed to transit the router.</p>

Note: Refer to the scenario entitled "Concurrent Routing and Source Route Bridging Connectivity Problems" in Chapter 2, "Connectivity Problem Scenarios," for illustrations and additional context-related information.

Routing in SRB Network Fails Unexpectedly

Symptom: As with the preceding situation, you may need to route certain protocols (for example, Novell IPX) through internetworks that are dominated by source route bridge (SRB) links. In this symptom, routing is working, then halts without any known administrative changes in the network.

Possible Causes and Suggested Actions

Table 4-2 outlines possible problems that could cause unexpected loss of communication over routers in SRB internetworks.

Table 4-2 Causes and Actions for Routing Failures in SRB Networks

Possible Cause	Suggested Actions
Software bug in the end station software	<p>Step 1: Enable debug token-ring on the router and examine the “Last Ring Status” line in the show interfaces display. If the status reads “Beaconing,” there is a problem with the ring.</p> <p>Step 2: If the “Last Ring Status” line does not indicate a beaconing state, check the output of the show rif EXEC command.</p> <p>Step 3: Determine whether the end station entry is missing from the RIF table.</p> <p>Step 4: If the end station is not listed, use the clear rif and clear arp commands, then ping the end station. See if the host can respond.</p> <p>Step 5: If the end station does not respond, use a network analyzer to look for the XID packet to NULL SAP (LSAP value of 00) sent by the router to the end station. The XID packet to NULL SAP is generated when the router’s RIF entry for a workstation ages out and the RIF table is being updated.</p> <p>Step 6: If you see the XID packet and the end station does not reply, there is probably a bug in the host software.</p> <p>Step 7: Upgrade your host software or contact your technical support representative for more assistance.</p> <p>Step 8: If you do not see the XID packet, or if the station replies, but you still cannot establish communication, contact your technical support representative.</p>

Note: Refer to the scenario entitled “Concurrent Routing and Source Route Bridging Connectivity Problems” in Chapter 2, “Connectivity Problem Scenarios,” for illustrations and additional context-related information.

No Communication over SRB

Symptom: Cisco routers can be configured to operate as source route bridges connecting two or more Token Rings. In this situation, the router is not forwarding SRB traffic.

Possible Causes and Suggested Actions

Table 4-3 outlines possible causes and recommended actions when routers are not bridging SRB traffic but were configured to do so.

Table 4-3 Causes and Actions for Blocked SRB Traffic

Possible Cause	Suggested Actions
Misconfigured Cisco router; ring number mismatch	Step 1: Get the ring number from IBM SRBs (specified in hexadecimal notation).
	Step 2: Check configuration with write terminal command. Look for ring number assigned to rings connected to router's interfaces. Ring numbers can be specified in either decimal or hexadecimal (with the 0xABCD notation).
	Step 3: Convert IBM SRB ring number to decimal and make sure ring numbers for all internetworking nodes agree.
	Step 4: If ring numbers do not agree, recalculate the router's ring number specification and make sure it matches the IBM SRB; reconfigure the router's interface specification.
	Step 5: If you still cannot communicate over the SRB, check subsequent symptom listings for possible causes. Contact your technical support representative if no actions restore communication.
End station does not support RIF	Step 1: Place a network analyzer on the same ring to which the end station is connected.
	Step 2: Look for any frames sent from the end station with the first bit of the source address set to 1. Refer to the scenario entitled "Translational Bridging, SRT, STUN, and SDLLC Connectivity Problems" in Chapter 2, "Connectivity Problem Scenarios," for illustrations and additional context-related information.
	Step 3: If no such frames are found, the end station does not support RIF and is not able to participate in source routing (example would be a UNIX or Novell workstation without source routing software).
	Step 4: Options: configure the router to act as an SRT; route the protocol if routable.
	Step 5: If your environment requires SRB, contact your workstation or server vendor for SRB drivers or for information about setting up SRB on your workstation or server.

Possible Cause	Suggested Actions
End station configured to send spanning explorers and router not configured to pass spanning explorers	<p><i>Step 1:</i> Place a network analyzer on the same ring to which the end station is connected.</p> <p><i>Step 2:</i> Look for any frames sent from the end station with the first bit of the source address set to 1.</p> <p><i>Step 3:</i> If such frames are found, check the first two bits of the routing control field and determine whether the frames are the spanning all ring frames (that is, first two bits are set to 1).</p> <p><i>Step 4:</i> If these frames are spanning all ring frames, determine whether the router is configured to forward spanning explorers (use source-bridge spanning interface subcommand).</p> <p><i>Step 5:</i> If necessary, add the source-bridge spanning interface subcommand to any router that is required to pass spanning explorers.</p> <p><i>Step 6:</i> If sessions still cannot be successfully established over the SRB, contact your technical support representative for more assistance.</p>

Blocked Communication over Remote SRB

Symptom: As a *remote* SRB, a router uses encapsulated Token Ring packets to allow interconnection of Token Ring networks over any non-Token Ring media type (such as an FDDI backbone, point-to-point serial lines, or a packet-switched network). In this situation, users/hosts are unable to communicate over the remote SRB.

Possible Causes and Suggested Actions

Table 4-4 outlines possible causes of being unable to communicate over remote SRB configurations and suggests actions to remedy these connection stoppages.

Table 4-4 Causes and Actions for Remote SRB Communication Problems

Possible Cause	Suggested Actions
Misconfigured source-bridge remote-peer commands on the router	<p>Step 1: Check the router configuration using the write terminal EXEC command.</p> <p>Step 2: Ensure that the source-bridge remote-peer command is pointing to the correct IP address on each router.</p> <p>Step 3: Modify configuration as required.</p> <p>Step 4: Check for existence of remote peers using the show source-bridge command.</p>
End station does not support RIF	<p>Step 1: Refer to “No Communication over SRB” earlier in this chapter.</p>
Hop count exceeded	<p>Step 1: Check the hop count values on the routers (using the show protocol route command) and other bridges in the path.</p>
No route to the remote peer (TCP/IP encapsulation)	<p>Step 1: Check the result of the show ip route command. If a route to the intended remote peer is not included in the list, create a route or check the state of devices and cabling in the path to the remote peer.</p> <p>Step 2: Verify IP connectivity; try to ping from the router to the remote peer IP address. If the remote peer does not reply, the SRB frames will not get through. If it does reply, IP routing is operational.</p>
Serial link problem	<p>Step 1: Make sure that the show interfaces command indicates that the serial port indicates the following: Serial <i>n</i> is up; line protocol is up. Refer to Chapter 7, “Troubleshooting WAN Connectivity,” if the status line indicates any other condition.</p> <p>Step 2: Make sure the selected encapsulation type matches the requirements of the network to which the serial interface is attached.</p>

Intermittent Communication Failures over Remote SRB

Symptom: Sessions time out over router configured for remote SRB.

Possible Causes and Suggested Actions

Table 4-5 outlines possible causes of intermittent connection when trying to pass traffic over a router configured for remote source route bridging (encapsulated SRB over any non-Token Ring media).

Table 4-5 Causes and Actions for Intermittent Connectivity over Remote SRB

Possible Cause	Suggested Actions
Sessions are timing out	<p>Step 1: Place a network analyzer on the ring local to the source station and look for acknowledgments appearing on the local ring after the transmission timeout period.</p> <p>Step 2: Check for dropped packets in the show interfaces display output on all involved interfaces in the path.</p> <p>Step 3: Check configuration for keyword local-ack at the end of the source-bridge remote-peer global configuration command.</p> <p>Step 4: Add this additional keyword if missing.</p> <p>Step 5: Adjust protocol parameters as described in Chapter 24 of the <i>Router Products Configuration and Reference</i> publication. In particular, the various LLC2 timer values may need tuning.</p>

Users Cannot Communicate over Cisco Translational Bridge

Symptom: Cisco routers allow for the translation between Ethernet and Token Ring of transparent bridging and source route bridging, respectively. Under certain circumstances, this translation may not work, resulting in an apparent failure of translational bridging.

Note: In certain situations, existing translational bridges have been replaced with Cisco translational bridges. This can cause interoperability problems. Some translational bridge implementations map functional addresses between media (such as LAT functional address 0900.2B00.00FA on Ethernet) to a broadcast address on the Token Ring ring side (such as C000.FFFF.FFFF). Cisco does not support this functionality. Furthermore, note that you cannot use translational bridging with any protocol that embeds a station's MAC address inside the information field of the MAC frames (examples include IP ARP and Novell IPX).

Possible Causes and Suggested Actions

Table 4-6 outlines possible causes that may prevent communication through routers configured for translational bridging.

Table 4-6 Causes and Actions for Traffic Stoppages over a Translational Bridge

Possible Cause	Suggested Actions
Router does not support Ethernet-to-Token Ring address mapping	<p>Step 1: Check for the existence of the Ethernet station using the show bridge EXEC command.</p> <p>Step 2: Use the show rif command to determine whether the target Token Ring station is visible on the internetwork.</p> <p>Step 3: If Ethernet and Token Ring end stations are visible, statically configure any relevant server MAC addresses in the client configurations, so the clients can listen to the server advertisements directly.</p> <p>(One case in which such a static mapping is required is when bridging DEC LAT traffic over a translational bridge. LAT services on Ethernet are advertised on a multicast address that is mapped by some translational bridges to a broadcast address on the Token Ring side. Cisco routers do not support this mapping.)</p>

Possible Cause	Suggested Actions
Vendor code mismatch	<p>Step 1: Specify ethernet-transit-oui global configuration command. This command forces the Cisco router to make the vendor code field 000000. This is frequently required when there are IBM 8209s (IBM Token Ring-to-Ethernet translating bridges) in the same network. Refer to the <i>Router Products Configuration and Reference</i> publication for more information about this command.</p> <p>(Older Token Ring implementations expect a vendor code (OUI field) of the SNAP header to be 000000. Cisco routers modify this field to be 0000F8 to specify that the frame was translated from Ethernet Version 2 to Token Ring.)</p>
Adding Cisco translational bridging destabilizes network, blocks all traffic	<p>Step 1: Check for pre-existing translational bridge(s) in parallel with the Cisco translational bridge; any that are left in place will result in loops.</p> <p>Step 2: Since implementing translational bridging defeats the spanning tree mechanism of both transparent bridging and SRB environments, you must eliminate all loops caused by insertion of the translational bridge.</p>
Trying to bridge protocols that embed MAC address in Information Field of MAC frame (such as IP ARP and IPX)	<p>Step 1: Route these protocols.</p> <p>Step 2: If you still cannot communicate over the router, contact your technical support representative.</p>

Traffic Cannot Get Through Router Implementing SRT

Symptom: Packets cannot traverse a router configured to support source-route transparent (SRT) bridging.

Note: Source-route transparent bridging allows you to implement transparent bridging in Token Ring environments. It is not a means to translate between SRB on a Token Ring and transparent bridging on Ethernet (or other) media. This confusion is sometimes the cause of blocked traffic in multimedia environments.

Possible Causes and Suggested Actions

Table 4-7 outlines possible causes and suggested actions to remedy blocked traffic flow in SRT implementations.

Table 4-7 Causes and Actions for SRT Communication Problems

Possible Cause	Suggested Actions
Trying to bridge frames containing RIF from Token Ring side to Ethernet side over SRT bridge	<p>Step 1: Use translational bridging instead of SRT to allow SRB-to-transparent bridging translation.</p> <p>Since SRT only performs transparent bridging between Ethernet and Token Ring, any packet containing RIF is dropped if SRT is being used.</p> <p>(When using translational bridging, the RIF of a packet received from the Token Ring side is extracted from the frame and saved in a table. The packet is then transmitted on the Ethernet side of the router. Later, the RIF is reinserted when a packet destined for the originating node is received by the router for transmission onto the Token Ring side of the router)</p>
Hardware does not support SRT	<p>Step 1: For each router interface required to support SRT, examine the output of the show interfaces token number command to determine whether the Token Ring interface is "Source Route Transparent capable."</p> <p>Step 2: Check all other bridges in network for SRT support.</p> <p>Step 3: Make sure that the software/microcode are compatible with SRT for all internetworking devices; upgrade as needed.</p>
Attempting to transfer large frame sizes (exceeding Ethernet MTU of 1500 bytes)	<p>Step 1: Configure hosts to generate frame sizes less than or equal to Ethernet MTU (1500 bytes).</p>
Trying to bridge protocols that embed MAC address in Information Field of MAC frame (such as IP ARP and IPX)	<p>Step 1: Route these protocols.</p> <p>Step 2: If you still cannot communicate over the router, contact your technical support representative.</p>

Users Cannot Make Connections over Router Configured for SDLLC

Symptom: Users cannot make session connections to hosts on the other side of a router configured to support SDLC-to-LLC Media Translation (SDLLC).

Possible Causes and Suggested Actions

Table 4-8 outlines possible causes and suggested actions when users are unable to make host connections over an SDLLC implementation.

Table 4-8 Causes and Actions for SDLLC Communication Problems

Possible Cause	Suggested Actions
Missing partner command	Step 1: Include partner interface subcommand (points router to the FEP's hardware address on Token Ring). This forces the transmission of explorer packets.
Missing sdllc xid command	Step 1: Include the sdllc xid interface subcommand. This command defines XID information (IDBLK and IDNUM) that must match host definitions when any 37X5 or 3172 device is being used as a gateway. Step 2: Check with host system administrators/programmers to ensure that XID information is properly defined.
Microcode incompatibility	Step 1: Perform show controller mci command to obtain microcode version of serial card. Look for the SCI microcode version. Step 2: Upgrade to the latest microcode version.
Incorrect RTS signal	Step 1: Insert a breakout box between the router and the IBM device, and monitor the LEDs for correct signaling. RS-232 signaling requirements are briefly described in the discussion following this table. Step 2: Check RTS for continuous active signal. Step 3: If the signal is not continuously active, set the signal high by strapping DTR from the IBM side to RTS on the router side. Open the RTS connection between the router and the IBM device. More information concerning physical layer mismatches is provided with a subsequent symptom module entitled "Router Is Not Communicating with IBM SDLC Devices over RS-232." Step 4: If the IBM device is a 3174, reply with a "1" to question number 340 in the 3174 configuration process (permanent request to send).
Incorrect V.35 applique jumper setting	Step 1: When using the V.35 dual-mode applique as a DCE, remove the SCT/SCTE jumper. This selects SCT (specifies timing signal from server).

IBM RS-232 Signaling Requirements Summary

When connecting a router to an IBM device with a serial connection, you must verify that the signaling configurations are compatible. Figure 4-1 illustrates a typical serial connection between a router (Router-1) and an IBM device. A breakout box is inserted to examine signal states on the cable.

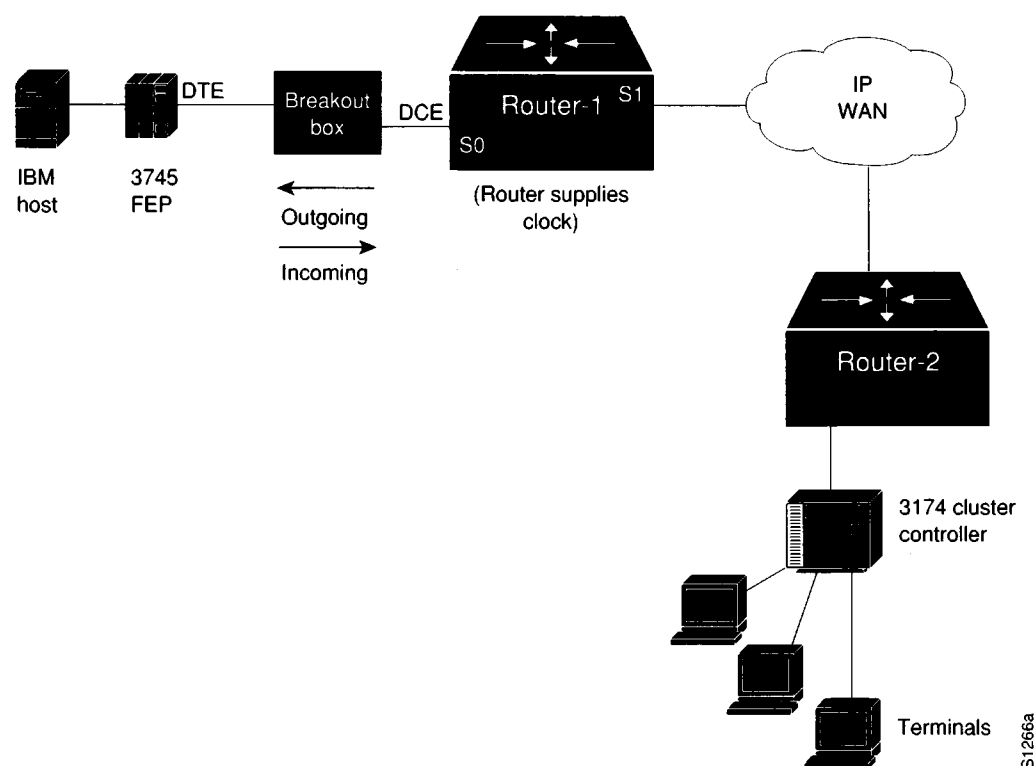


Figure 4-1 Checking IBM Serial Link to Router with Breakout Box

Table 4-9 outlines the key signaling requirements for the RS-232 link between Router-1 and the 3745 FEP. Figure 4-1 illustrates the direction of signals with respect to the router as listed in Table 4-9. This environment assumes that the router is configured for DCE, while the IBM FEP is configured for DTE.

Table 4-9 Key RS-232 Signaling Requirements for Router to IBM FEP Connection

Lead/Signal	State	Reference to Router
4/RTS	High	Incoming
5/CTS	High	Outgoing
6/DSR	High	Outgoing
8/Carrier Detect	High	Outgoing
20/DTR	High	Incoming

Preventive Actions in SDLLC Environments

When configuring a Cisco router for SDLLC operation in IBM internetworking environments, the following actions are suggested for preventing operational problems.

1. When configuring SDLLC, the **sdllc traddr** command must point to the virtual ring, not to the physical ring. The virtual ring corresponds to the ring group number specified in the **source-bridge ring-group** command. This applies to single router configurations (where the Token Ring and the serial line are both tied to the same router) and multirouter configurations (where the routers are separated by WAN clouds). Also note that the specification of the virtual ring number is the last parameter to be included in the **sdllc taddr** command.
2. SDLLC will not work between an IBM AS/400 and 5394. Do not try to use Cisco's SDLLC feature to provide this functionality. The AS/400 can only operate as a PU 2 device, while the 5394 can only operate as a PU 1 device. SDLLC only accommodates protocol and frame translation at the DLC level and does not participate in any SNA level exchange. To allow for this kind of translation, you must implement some kind of conversion device for translating PU 1 to PU 2.

Note: Refer to the *Router Products Configuration and Reference* publication for more information concerning these commands.

Virtual Token Ring Addresses and SDLLC Implementations

The **sdllc traddr** command requires that you specify a virtual Token Ring address for an SDLC-attached device (the device that you are spoofing to look like a Token Ring device). The last two hex digits of this virtual ring address must be 00 (hex). This is because the last byte of the address represents the SDLC address of the station on the serial link.

Use care in assigning virtual ring addresses. Any virtual ring address that falls into the range xxxx.xxxx.xx00 to xxxx.xxxx.xxFF belongs to the associated SDLLC serial interface. An IBM Locally Administered Address (LAA) is typically user-defined, and in practice these addresses tend to follow a logical ordering. As a result, there is a real chance that other IBM devices on an internet will have an LAA that falls in the same range. If this occurs, problems can arise because Cisco routers only examine the first 10 digits of the LAA address of a packet (not the last two, which are considered wild cards). If the router sees a match of the assigned SDLLC LAA address, it automatically forwards that packet to the SDLLC process. In certain cases, this can result in packets being lost (incorrectly forwarded to the SDLLC process) and sessions never being established.

Note: Before assigning a virtual ring address for any SDLLC implementation, be sure to obtain the LAA naming convention used in the internet to avoid conflicting address assignments.

Intermittent Connectivity over Router Configured for SDLC

Symptom: User connections to hosts time out over a router configured to support SDLC Transport.

Possible Causes and Suggested Actions

Table 4-10 outlines possible causes and suggested actions when connection service availability is erratic to hosts over an SDLC implementation.

Table 4-10 Causes and Actions for Intermittent SDLC Connectivity

Possible Cause	Suggested Actions
SDLC timing problems	<p>Step 1: Place a serial analyzer on the serial line attached to the source station and monitor packets.</p> <p>Step 2: If duplicates appear, check configuration for keyword local-ack at the end of the stun route address interface subcommand.</p> <p>Step 3: Add this additional keyword if missing on both router configurations for SDLC interfaces.</p> <p>Step 4: Adjust SDLC protocol parameter described in the <i>Router Products Configuration and Reference</i> publication. These parameters are used to customize SDLC transport over various network configurations. In particular, the various LLC2 timer values may need tuning.</p>

Router Is Unable to Connect to Token Ring

Symptom: When installing a new router in a Token Ring environment, you find that the router will not connect to the ring.

Possible Causes and Suggested Actions

Table 4-11 outlines possible causes and suggested actions when the router fails to connect to the Token Ring.

Table 4-11 Causes and Actions for Router Not Connecting to Ring

Possible Cause	Suggested Actions
Relay open in MAU	<p>Step 1: At system power-on, if “open lobe fault” message appears on console (or VTY) display connected to router, check the cable connection to MAU.</p> <p>Step 2: Issue the clear interface command to reset the Token Ring interface (used to clear the interface and reinsert into the ring). For all Token Ring cards except the CTR and low-end boxes, you must use the clear interface command to reinitialize the Token Ring interface if the interface was down.</p> <p>Step 3: Issue a show interfaces token command to verify interface operation.</p> <p>Step 4: If the interface is operational, but “open lobe fault” message persists and router continues to be unable to connect to its ring, switch connection from router to a different MAU port.</p> <p>Step 5: If “open lobe fault” message continues to appear, disconnect all devices from the MAU and reset the MAU’s relay with the tool provided by MAU vendor.</p> <p>Step 6: Reattach the router and determine whether it can connect to the ring. If resetting the relay does not remedy the problem, try replacing the MAU with one that is known to be operational.</p> <p>Step 7: If the router is still unable to connect to the ring, check internal cable connections of router Token Ring cards. Ensure that cables associated with the respective port numbers and applique numbers are correctly wired (make sure they are not swapped).</p> <p>Step 8: If router still cannot connect to the Token Ring, replace the cables connecting the router to the MAU with working cables.</p> <p>Step 9: Use the clear interface command to reset interface and reinsert the router into the ring. Using the show interfaces token command, look for “interface up” and “protocol up” in the status line.</p>

Possible Cause	Suggested Actions
Bad ring speed specification	<p><i>Step 1:</i> Use show interfaces token command to determine status of interface.</p> <p><i>Step 2:</i> If status line indicates that the interface and line protocol are not up, check cable from router to the MAU. Make sure that the cable is good; replace if necessary.</p> <p><i>Step 3:</i> If show interfaces token indicates interface up/line protocol up, use the ping command between routers to test connectivity.</p> <p><i>Step 4:</i> If the remote router does not respond, check the ring specification on all nodes attached to the Token Ring backbone. Ring speed for all must be the same.</p> <p><i>Step 5:</i> If necessary, modify ring speed specifications for clients, servers, and routers.</p> <p><i>Step 6:</i> Use the ring speed command to modify ring speed configuration for IGS/TR. Change jumpers as needed for modular router platforms. Refer to your system's hardware installation and maintenance manual for more information about ring speed specification.</p>

Router Is Not Communicating with IBM SDLC Devices over RS-232

Symptom: When installing a router, you find that the router is not able to communicate with an IBM SDLC device over an RS-232 cable.

Note: When debugging serial line physical layer problems, it is important to observe indicator lights on appliques, LEDs on modems/modem eliminators, and line drivers. These will help determine if the hardware is having any problems and can save time in debugging other problems.

Possible Causes and Suggested Actions

Table 4-12 outlines possible causes and suggested actions when a router is apparently not communicating with SDLC devices over RS-232.

Table 4-12 Causes and Actions for Blocked Communication with SDLC Device

Possible Cause	Suggested Actions
Physical layer mismatch	<p>Step 1: Make sure that both the IBM device and the router implement the correct signal coding (NRZ or NRZI).</p> <p>Step 2: If the IBM device supports full-duplex NRZ, make sure it is set for full duplex NRZ (set RTS high). Set the signal high by strapping DTR from the IBM side to RTS on the Cisco side.</p> <p>Step 3: For AS/400 multidrop devices, make sure that the serial line connecting the router with the primary link station has carrier detect tied to ground.</p> <p>Step 4: Use the show interfaces command to determine whether the interface and protocol are up.</p> <p>Step 5: If the router is set up as a DTE device, check to make sure the clocking source configurations match for all devices. Make sure the modems/modem eliminators are properly configured.</p> <p>When installing routers into IBM environments, make sure that the IBM devices are properly configured to communicate with each other. For example, make sure that cluster controllers can talk to FEPs before adding a router.</p> <p>Make sure that the correct clock rate is set to match the network's (externally derived) clock.</p> <p>Step 6: If the clock settings match, try reducing the line speed to 9600 baud (whether the router is configured as DTE or DCE).</p>

SDLC Sessions Fail over Router Running STUN

Symptom: SDLC sessions are not coming up when attempted over a router running STUN between two nodes. An underlying symptom here is that the SDLC sessions are not being completed—the necessary handshaking is not occurring.

Possible Causes and Suggested Actions

Table 4-13 outlines possible causes and suggested actions when SDLC sessions are not coming up over a router implementing STUN.

Table 4-13 Causes and Actions when SDLC Sessions Fail over STUN

Possible Cause	Suggested Actions
Broken physical connectivity of SDLC secondary stations and the Cisco STUN peer	<p>Step 1: Check the STUN state with the show stun command from the router.</p> <p>Step 2: If STUN state indicated using the show stun command is “closed,” check physical connectivity as described in the previous symptom/problem discussion, “Router Is Not Communicating with IBM SDLC Devices over RS-232.”</p>
Misconfigured stun route address command specification	<p>Step 1: If STUN state is “open,” use the debug stun-packet command at the router(s) to look for Set Normal Response Mode (SNRM) and matching Unnumbered Acknowledgment (UA) packets. Ensure that the SNRMs and UAs (that have SDLC addresses corresponding to the relevant secondary stations) are getting to the correct STUN peer (router).</p> <p>Step 2: If SNRMs are detected with the debug stun-packet command output, but no UAs are returning, use the write terminal command on the router to which the primary link station is attached.</p> <p>Look for the SDLC address specified in the stun route address command. Entries for this command should point to relevant secondary link stations.</p>
Misconfigured stun peer command	<p>Step 1: At the STUN peer (router) to which the secondary link station is attached, enable debug stun-packet and look for SNRMs for that peer.</p> <p>Step 2: If no SNRMs appear in the debug output, check the stun peer-name commands on the router to which the primary link station is attached. Make sure that the IP address specified in this command is correct for the router.</p>

Possible Cause	Suggested Actions
Physical connectivity problem from the secondary link station to the router; misconfigured stun route address definition on router to which the secondary link station is attached; or broken IBM gear	<p>Step 1: If you do see SNRMs, use the show interface serial command to see if output drops are accumulating. This would suggest that the router is not communicating with the secondary link station.</p> <p>Step 2: For 3174s, if no output drops are detected, check the front panel display for values cycling between 532 and 505. If the 3174 display is cycling between 532 and 505, then communication from the router to the secondary link station works. This cycling of values indicates that SNRMs are getting to the 3174, but the receiver ready signal is not initializing.</p> <p>Step 3: Check the output of the debug stun-packet display to see if relevant UAs are being detected. If so, the problems of physical connectivity and broken IBM gear can be eliminated.</p> <p>If debug stun-packet output (at router to which the primary link station is attached) displays relevant UAs, the problem is isolated to a physical connectivity problem from that router to the primary link station.</p> <p>Step 4: Check physical connectivity as described in the previous symptom/problem discussion, "Router Is Not Communicating with IBM SDLC Devices over RS-232."</p>

NetBIOS Devices Cannot Communicate over Remote SRB

Symptom: Users on NetBIOS clients complain that they cannot establish connections to NetBIOS servers over routers acting as remote SRBs.

Possible Causes and Suggested Actions

Table 4-14 outlines possible causes and suggested actions when access is blocked to NetBIOS servers from clients over a remote SRB cloud.

Table 4-14 Causes and Actions for Blocked NetBIOS Communication

Possible Cause	Suggested Actions
Incorrect mapping of NetBIOS name cache server-to-client mapping	<p>Step 1: For each router on which NetBIOS name caching is enabled, use the show rif command to determine whether the RIF entry shows the correct path from the router to <i>both</i> the client and the server.</p> <p>Step 2: Use the show netbios command to see if the NetBIOS cache entry show the correct mapping from server name and client name to MAC address.</p> <p>Step 3: Use the write terminal command at each router and examine the mapping of addresses specified in the netbios name-cache command. Change these if they are not correct.</p>
Incorrect specification of remote peer parameters in source-bridge specification	<p>Step 1: For each router on which the name cache is enabled, use the show source-bridge command to obtain the <i>version</i> of the remote connection. The value specified should 2.</p> <p>Step 2: If the value is 1, connections will not get through and you must modify your configuration. Specify the version 2 parameter of the source-bridge remote-peer command.</p>

Note: Whenever name caching appears not to be running between a particular client and server, capture traces for packets that apparently are not flowing. In addition, get the output of **show rif**, **show netbios**, and **show source** commands for the routers at each end of the RSRB cloud. This information will help in diagnosing a NetBIOS name caching problem by providing information about the state of the router.

Router Cannot Be Linked from LAN Network Manager

Symptom: A specific router cannot be linked from LAN Network Manager (LNM) in an SRB environment.

Possible Causes and Suggested Actions

Table 4-15 outlines possible causes and suggested actions when a router cannot be linked using LNM.

Table 4-15 Causes and Actions for Problems Linking Router via LNM

Possible Cause	Suggested Actions
Misconfigured LAN Network Manager MAC address specifications (universal)	<p>Step 1: Use the show lnm config command on the router to determine the Token Ring MAC addresses. They must match the addresses entered on the LNM.</p> <p>Step 2: If they do not match, enter these Token Ring MAC addresses on the LNM platform.</p>
MAC address mismatch when router is connected to a virtual ring (locally administered)	<p>Step 1: Use the show lnm config command on the router to determine the Token Ring MAC addresses.</p> <p>Step 2: Make sure that the Token Ring address configured on the LNM matches the address administered on the router. Check the mac-address command for each Token Ring interface. This command gives each Token Ring interface a locally administered address (such as 4000.0001.2345).</p>

Chapter 5

Troubleshooting Novell Connectivity

5

Novell IPX Internet Diagnostic Overview 5-1

Problem Isolation in Novell IPX Networks 5-2

Novell Internetworking Connectivity Symptoms 5-3

Symptom Summary 5-3

Clients Cannot Communicate with NetWare Servers over Router 5-4

Possible Causes and Suggested Actions 5-4

SAP Updates Not Getting Through Router 5-9

Possible Causes and Suggested Actions 5-9

Novell NetBIOS Packets Cannot Get Through Router 5-11

Possible Causes and Suggested Actions 5-11

Helper Address Specification Hints 5-13

Basic Helper Address Assignment 5-13

Helper Address Configuration over Single Serial Interconnection 5-14

Helper Address Configuration over Multiple Serial Interconnections 5-15

Helper Address Configuration for Reverse Broadcast NetBIOS Servers 5-17

Helper Behavior with Parallel Routers 5-18

Clients Cannot Connect to Server over Packet-Switched Network 5-21

Possible Causes and Suggested Actions 5-21

Notes About PSN Address Map Specifications 5-22

Address Mapping for Novell-to-X.25 Interconnection 5-22

Address Mapping for Novell-to-Frame Relay Interconnection 5-23

Chapter 5

Troubleshooting Novell Connectivity

5

This chapter presents protocol-related troubleshooting information for Novell IPX connectivity problems. The emphasis here is on symptoms and problems associated with IPX network connectivity.

This chapter consists of the following sections:

- Diagnostic tools and techniques used for isolating Novell internetworking problems
- Overview of symptom list
- Symptom/problem/action modules

The problem/solution modules consist of the following sections:

- Symptom statement—A specific symptom, associated with the technology/media/protocol in which this module appears.
- Possible causes and suggested actions—For each symptom, a table containing possible causes for the symptom and suggested actions for resolving each cause.

Novell IPX Internet Diagnostic Overview

The following tools (all available via the router's basic management interface) form the core of the administrator's internetwork toolkit:

- **ping** command—The **ping** command is essential in determining where failures are occurring. However, this tool is only useful for determining connectivity between routers, because it is not supported by Novell servers.
- **show** commands—The **show** commands provide information about interface conditions, protocol status, neighbor reachability, router configuration and status, level of traffic, errors and drops, and other network data. In particular, **show novell servers** and **show novell route** can be used to help build a topology map and determine connectivity among network entities.

In addition, the **show interfaces [ethernet | fddi | serial | token] number** command provides essential information concerning interface and media status. Remember to include the interface number when using this form of **show interfaces**.

- **debug** commands—The **debug** commands provide clues about the status of both routers and other network entities. Use **debug** commands with great care. Using the wrong **debug** at the wrong time can exacerbate problems in poorly performing networks. Of particular usefulness is the **debug novell-packet** command.



Caution: Throughout this publication, the use of **debug** commands is suggested for obtaining information about network traffic and router status. Use these commands with great care. In general, it is recommended that these commands only be used under the direction of your router technical support representative when troubleshooting specific problems. Enabling debugging can disrupt operation of the router when internets are experiencing high load conditions. When you finish using a **debug** command, remember to disable it with its specific **undebug** command or with the **undebug all** command.

When specific diagnostic commands are considered particularly useful for troubleshooting symptoms, they are listed with the specific symptom discussion in this chapter.

Refer to “Using Cisco Diagnostic Tools” in Chapter 1 and to Chapter 10, “Debug Command Reference,” for more information about using these tools. In addition, details about other specific commands are provided in the *Router Products Configuration and Reference* publication.

Problem Isolation in Novell IPX Networks

One important consideration to remember when troubleshooting broken interconnections is that normally everything does not break at the same time. As a result, you can typically work out from an operational node to the point of failure. The following basic steps will help when trying to isolate the source of connection disruption:

- Step 1:** First, determine whether your local host is properly configured.
- Step 2:** Next, use the **ping EXEC** command to determine whether the routers through which you must communicate can respond. Start with the most local router and progressively “ping out” through the internet.
- Step 3:** If you cannot get through a particular node, examine the node’s configuration if possible and use the various **show** commands to determine the router’s state. In particular, **show novell traffic** and **show novell interface** can provide useful information.
- Step 4:** If you can get to all the routers in the path, check the host configuration at the remote host (or get someone’s help to do so) and check its configuration.
- Step 5:** Check the routers’ routing tables (**show novell route**) and table of servers (**show novell servers**) for any anomalies (such as duplicate network numbers) and to see whether the host(s) in question are appearing.

Novell Internetworking Connectivity Symptoms

The symptom modules that follow pertain to Novell internetwork problems. Unless otherwise indicated, each module is presented as a set of general problems. Where there are special considerations associated with a situation, notes are included.

Symptom Summary

The following Novell internetworking connectivity symptoms are discussed in this section:

- Clients cannot communicate with NetWare servers over routers.
- SAP updates are not getting through router.
- Novell NetBIOS packets cannot get through router.
- Clients cannot access server over packet-switched network.

Note: For the puposes of this document, symptoms, problems, and actions associated with Novell NetWare 2.15 apply equally to NetWare 2.2, unless NetWare 2.2 is specifically excluded.

Clients Cannot Communicate with NetWare Servers over Router

Symptom: Clients may or may not be able to connect to servers on their directly connected networks. In either case, no connections can be made to servers on the other side of the router.

Possible Causes and Suggested Actions

Table 5-1 outlines possible causes of blocked access to servers over routers.

Table 5-1 Causes and Actions for Blocked NetWare Connectivity over Router

Possible Cause	Suggested Actions
Clients or servers are not attached to network	<p>Step 1: Connect both clients and servers to the same network and verify that they can communicate.</p> <p>Step 2: If they cannot communicate, check configuration of the client and server. Refer to host software documentation for troubleshooting information.</p> <p>Step 3: Attach a network analyzer to the network to which clients and servers are temporarily connected. Look for source addresses of both.</p> <p>Step 4: If you find the source addresses, end stations are operating properly. If you do not find their addresses, check the configuration of the clients and servers (consult your client/server documentation for more information).</p>
Router interface is not functioning	<p>Step 1: Check operation of router using show interfaces command. Look for interface and line protocol as “up” in status line.</p> <p>Step 2: If the interface is “administratively down,” specify the no shutdown command on the interface.</p> <p>Step 3: If either the interface or protocol is down, check cable connections from the router. If necessary, replace the cable.</p> <p>Step 4: If after replacing the cable, you still cannot get the interface to come up (interface and line protocol shown as “up” in show interfaces output), contact your router technical support representative.</p>

Possible Cause	Suggested Actions
Router network number specification is misconfigured for NetWare 2.15, causing problems for RIP	<p>Step 1: Check router configuration to see whether Novell routing is enabled. If not, add the novell routing global configuration command and other commands as necessary (refer to <i>Router Products Configuration and Reference</i> publication).</p> <p>Step 2: Get the network number from target network server.</p> <p>Step 3: Compare with network number specified on router (using write terminal or show novell interface commands).</p> <p>Step 4: If network numbers do not match, reconfigure router with correct network numbers.</p> <p>Step 5: If network numbers do match, check the router interface on the client side and make sure that the network number assigned is unique with respect to all network numbers in your Novell internetwork. On the server side of the router, make sure that the network number assigned to the router interface matches the network number for the server.</p>
Router network number specification is misconfigured for NetWare 3.11, causing problems for RIP	<p>Step 1: Check router configuration to see whether Novell routing is enabled. If not, add the novell routing global configuration command and other commands as necessary (refer to <i>Router Products Configuration and Reference</i> publication).</p> <p>Step 2: Get the <i>external</i> network number of the server's interface attached to the network to which the router is also attached. Do not use the 3.11 server's <i>internal</i> network number.</p> <p>Step 3: Compare the <i>external</i> network number with network number specified on router (write terminal or show novell interface commands).</p> <p>Step 4: If network numbers do not match, reconfigure the router with correct network numbers.</p> <p>Step 5: If network numbers do match, check the router interface on the client side and make sure that the network number assigned is unique with respect to all network numbers in your Novell internetwork.</p>
NetWare 2.15 and 3.11 network number mismatch on same network or backbone. This causes problems for RIP, which relies on network numbers to route traffic	<p>Step 1: If NetWare 2.15 servers are on the same physical cable with NetWare 3.11 servers, the network number for the connected interface of any 2.15 server and the external network number for the connected interface of any 3.11 server must match. Compare the <i>external</i> network numbers for the 3.11 servers with the network numbers for the 2.15 servers.</p> <p>Step 2: If these numbers do not match, reconfigure the servers to make them match. Refer to the server documentation for information concerning these modifications.</p>

Possible Cause	Suggested Actions
Bad access list	<p>Step 1: Remove novell access-group specifications on all relevant interfaces.</p> <p>Step 2: See whether traffic can get through by testing the connection function from the client to the target server.</p> <p>Step 3: If connection now works, access list needs modification.</p> <p>Step 4: Apply one access list statement at a time until you can no longer create connections to isolate the location of the bad access list specification.</p> <p>Step 5: Make sure that access lists are applied to the correct interface. Filters must be applied to the <i>outgoing</i> interfaces for normal (standard) traffic filters.</p>
Backdoor bridge between segments	<p>Step 1: Use the show novell traffic command and determine whether the “bad hop count” field is incrementing.</p> <p>Step 2: If this counter is increasing, use a network analyzer to look for packet loops on suspect segments. Look for routing and SAP updates. If a backdoor bridge exists, you are likely to see hop counts incrementing up to 16; the route will disappear and can reappear unpredictably.</p> <p>Step 3: Also look for known <i>remote</i> network numbers that show up on the <i>local</i> network. That is, look for remote network numbers that are not being handled by the router (source address is not the router’s).</p> <p>Step 4: Isolate the local Ethernet into smaller segments (using a fanout or similar device).</p> <p>Step 5: Examine the traffic on each segment with a protocol analyzer. If a packet appears from a known remote node (has a remote source address) the back door is located on that segment.</p>
Duplicate network numbers on Novell servers	<p>Step 1: Use the show novell servers command to look for duplicate network numbers.</p> <p>Display output generates list of servers by name, network number, MAC address, hop count, and interface.</p> <p>Step 2: If duplicates are found, modify server configurations so no duplicates exist on your internet.</p>

Possible Cause	Suggested Actions
Nonfunctional FDDI ring	<p>Step 1: Use the show interfaces fddi command to determine status of interface.</p> <p>Step 2: If show interfaces fddi indicates interface up/line protocol up, use the ping novell command between routers to test connectivity to routers.</p> <p>Step 3: If interface is up and line protocol is up, make sure the MAC addresses of upstream and downstream neighbors is as expected. If all zeros appear in either of the address fields for these neighbors, a physical connection problem is likely.</p> <p>Step 4: In this case, (or if status line does <i>not</i> indicate interface up/line protocol up), check connections at patch panel or connectivity between routers using an optical TDR or light meter. Ensure that signal strength is within specification.</p>
Nonfunctional serial link	<p>Step 1: Use show interfaces serial command to determine status of interface.</p> <p>Step 2: If show interfaces serial indicates interface up/line protocol up, use the ping novell command between routers to test connectivity to routers.</p> <p>Step 3: If routers do not respond to ping test, follow troubleshooting techniques as discussed in Chapter 7, "Troubleshooting WAN Connectivity."</p>
Nonfunctional Ethernet backbone	<p>Step 1: Use the show interfaces ethernet command to determine status of the interface. Determine whether status indicates interface up/line protocol up.</p> <p>Step 2: If the status line does not indicate interface up/line protocol up, check the router's physical attachment to Ethernet backbone.</p> <p>Step 3: If show interfaces ethernet indicates interface up/line protocol up, use the ping novell command between routers to test connectivity to routers.</p> <p>Step 4: Obtain analyzer traces and look for packets from target servers, clients and routers.</p> <p>Step 5: Any known nodes that do not appear as expected are suspects for being problem nodes. Locate and determine whether entity and attached cables are functional. If not, replace or reconfigure as needed.</p>

Possible Cause	Suggested Actions
Mismatched Ethernet encapsulation methods	<p>Step 1: Check encapsulation types for clients and servers.</p> <p>Step 2: Compare with router encapsulation type assigned. (By default, Cisco routers use Novell's Frame Type Ethernet_802.3 encapsulation. Cisco refers to this as "novell-ether" encapsulation)</p> <p>Step 3: If servers and client are using what Novell refers to as "Frame Type Ethernet _II," make sure that the router is also using this form—assigned using the novell encapsulation arpa interface subcommand. (This particular encapsulation mismatch problem would apply to DEC/VMS hosts/servers running Novell server software).</p> <p>Step 4: If clients and servers are using SNAP or Ethernet_802.2 encapsulation, change client/server encapsulation type to either Frame Type Ethernet_802.3 or Ethernet_II.</p> <p>Step 5: As a last resort, disable Novell routing and enable bridging. (Cisco routers do not support SNAP or Ethernet_802.2 encapsulation of Novell IPX at this time, except SNAP on Token Ring.) Note that Cisco routers can translate Frame Type Ethernet_802.3 and Frame Type Ethernet_II encapsulation types between different interfaces. Each interface must use a different network number.</p>
Nonfunctional Token Ring backbone	<p>Step 1: Use show interfaces token command to determine status of interface.</p> <p>Step 2: If status line indicates that the interface and line protocol are not up, check cable from router to the MAU. Make sure that the cable is good; replace if necessary.</p> <p>Step 3: If show interfaces token indicates interface up/line protocol up, use the ping novell command between routers to test connectivity to them.</p> <p>Step 4: If the remote router does not respond, check the ring specification on all nodes attached to the Token Ring backbone. Ring speed for all must be the same.</p> <p>Step 5: If necessary, modify ring speed specifications for clients, servers, and routers.</p> <p>Step 6: Use the ring speed command to modify ring speed configuration for IGS/TR. Change jumpers as needed for modular router platforms. Refer to your system's hardware installation and maintenance manual for more information about ring speed specification.</p>

SAP Updates Not Getting Through Router

Symptom: Service Advertisement Protocol (SAP) packets generated by servers broadcast the specific Novell services offered by a particular server. Here, the SAP updates appear to be unable to get through from one side of a router to the other.

Possible Causes and Suggested Actions

Table 5-2 outlines possible causes of SAP updates not getting broadcast on the other side of a router.

Table 5-2 Causes and Actions for SAP Updates Not Being Broadcast

Possible Cause	Suggested Actions
Novell server is not sending SAP updates	<p>Step 1: Use protocol analyzer to look for SAP updates from the server.</p> <p>Step 2: If the server is not sending SAP updates, make sure the server is attached to the networks and that it is configured to send SAP updates (not configured to withhold SAP updates).</p> <p>Step 3: In an Ethernet-based environment, if the server is sending SAP updates, check the encapsulation type in the router configuration; it must match the Novell server encapsulation specification (Frame Type Ethernet_802.3 or Frame Type Ethernet_II).</p>
Ring speed specification discrepancy	<p>Step 1: Check ring speed specifications on Novell servers and routers (4 or 16 Mbps).</p> <p>Step 2: If they do not match, for IGS/TR, use ring speed command to make the router configuration match server specifications. For modular systems, refer to the appropriate installation and maintenance manual for proper 4- or 16-Mbps jumper settings.</p>
Bad access lists	<p>Step 1: Disable any SAP-specific access lists by removing novell input-sap-filter and novell output-sap-filter commands as appropriate.</p> <p>Step 2: If you have a Novell client on other side of router, use the client's slist command to verify that services are being advertised for the server. The display servers command can also be used from the server console.</p> <p>Step 3: As an alternative, run debug novell-sap command on the router. Look for server name, network number, and MAC address.</p> <p>Step 4: If the Novell server's SAP information is included in the router's updates, the access list is causing SAP updates to be dropped at the router.</p> <p>Step 5: Revise access lists or filter statements as necessary and apply individually to ensure that updates are being distributed appropriately.</p>

Possible Cause	Suggested Actions
Misconfigured network number on router or Novell server. This causes problems for RIP, which relies on network numbers to route traffic.	<p>Step 1: Verify that there are no duplicate network numbers. Each Novell network must be unique.</p> <p>If the routers or Novell servers have duplicate network numbers, the router might not send out SAP updates.</p> <p>Step 2: Use show novell route and/or show novell servers to determine whether any network number is duplicated on your network.</p> <p>Step 3: Ensure that all network numbers are unique. If not, modify server configurations or router's novell network specification as appropriate.</p>
Novell servers unable to handle the rate at which routers generate SAP updates	<p>Step 1: Compare output of show novell servers from the router with output of slist command from Novell clients.</p> <p>Step 2: If the slist output on Novell server shows a partial listing of SAP entries, this is the likely problem.</p> <p>Step 3: To remedy, use the novell output-sap-delay command to specify the delay between packets in a multipacket SAP update. Use the lowest possible delay that corrects the problem. An initial value of 5 msec is recommended.</p>
Server configured to withhold SAP updates	<p>Step 1: Examine the specific server's configuration. Consult with the server's documentation to determine how to find this information.</p> <p>Step 2: If server is set to withhold SAP updates, change configuration to allow SAP updates.</p> <p>Step 3: Check for connectivity between file server and router, using the show novell server command.</p>
Limited-user version of NetWare software	<p>Step 1: Check the copy of software running on the server. If it is a limited-user version, you must upgrade the version to support more users.</p>
Nonunique MAC address on routers	<p>Step 1: Use the write terminal command to examine the current configuration of each router in the path.</p> <p>Step 2: Check the hardware address specified in the novell routing global configuration command.</p> <p>Step 3: If this system-generated number matches for both routers, reinitialize one of the routers and see whether connectivity over the link is re-established.</p> <p>Step 4: If the numbers still match, then get a real MAC address from one of the interfaces using the show interfaces command and assign that address to the router using the novell routing command (example: novell routing 0000.3080.09a7).</p> <p>In general, this problem occurs more frequently in Token Ring implementations.</p>

Novell NetBIOS Packets Cannot Get Through Router

Symptom: Clients are unable to get response from servers running Novell NetBIOS when connections are attempted over a router.

Possible Causes and Suggested Actions

Table 5-3 outlines possible causes of clients' inability to connect to Novell NetBIOS servers over a router.

Table 5-3 Causes and Actions for Blocked NetBIOS Traffic

Possible Cause	Suggested Actions
Missing novell helper-address command	<p>Step 1: Run the debug novell-packet command. Look for Novell packets with unknown type 20 specification.</p> <p>Step 2: Use write terminal command to check router configuration for novell helper-address interface subcommand configured for incoming interface (for Novell NetBIOS traffic from stations).</p> <p>Step 3: If the novell helper-address command is not present, add it as appropriate.</p> <p>Depending on the network configuration, the specification of the helper address may differ.</p> <p>Refer to the "Helper Address Specification Hints" following this table for suggestions and to the <i>Router Products Configuration and Reference</i> publication for more information.</p>
Misconfigured novell helper-address specification	<p>Step 1: Check router interface configuration on client side for novell helper-address specification.</p> <p>Step 2: Make sure the MAC address field specified in this command is a type of broadcast.</p> <p>All nets broadcast example:</p> <pre>interface ethernet 0 novell helper-address -1.ffff.ffff.ffff</pre> <p>Directed broadcast example:</p> <pre>interface ethernet 1 novell helper-address a4.ffff.ffff.ffff</pre>

Possible Cause	Suggested Actions
Bad access list	<p><i>Step 1:</i> Remove novell helper-list specifications on all relevant interfaces.</p> <p><i>Step 2:</i> See whether traffic can get through by testing the connection function from the client to the target server.</p> <p><i>Step 3:</i> If connection now works, access list needs modification.</p> <p><i>Step 4:</i> Apply one access list statement at a time until you can no longer create connections to isolate the location of the bad access list specification.</p> <p><i>Step 5:</i> Make sure that access lists are applied to the correct interface. Helper lists are applied to <i>incoming</i> interfaces.</p>

Helper Address Specification Hints

Your *Router Products Configuration and Reference* publication provides details about configuration commands associated with helper address assignment. Refer to “Routing Novell IPX” in that publication for more information. The following illustrations and accompanying text discuss some of the implications of Novell helper address assignment, some potential pitfalls, and general behavioral characteristics.

Basic Helper Address Assignment

Consider the simple configuration illustrated in Figure 5-1. In this case, a router (Router-A) separates two Ethernet segments (Ethernet-2a and Ethernet-1d). Clients on Ethernet-2a must be able to access services from Server-1 and Server-2 on Ethernet-1d.

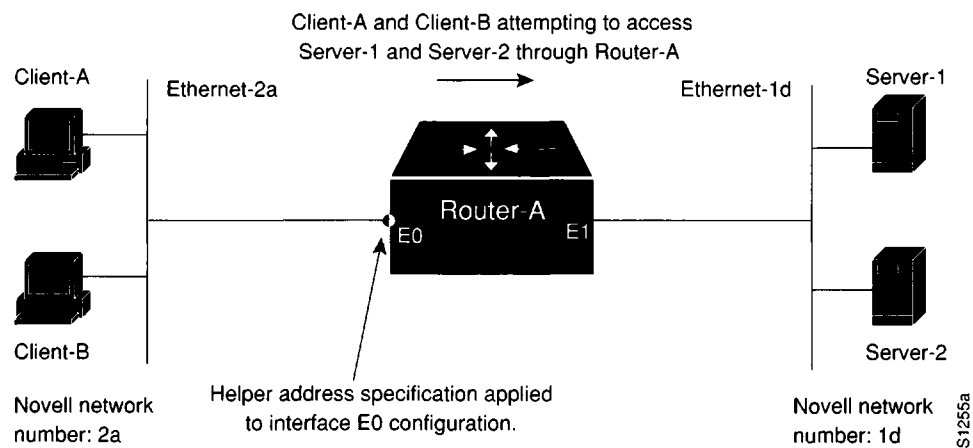


Figure 5-1 Basic Helper Address Network

The helper address specification would be as follows:

```
interface ethernet 0
novell network 2a
novell helper address -1.ffff.ffff.ffff
```

Here, -1 is used to specify flooding to all nets.

As an alternative, you also can specify a specific network number. In this case, you would specify Novell network 1d as follows:

```
interface ethernet 0
novell network 2a
novell helper address 1d.ffff.ffff.ffff
```

Helper Address Configuration over Single Serial Interconnection

The configuration illustrated in Figure 5-2 is nearly identical to the configuration illustrated in the preceding illustration, except that here Ethernet-2a and Ethernet-1d are separated by a serial network and two routers (Router-A and Router-B). As before, clients on Ethernet-2a must be able to access services from Server-1 and Server-2 on Ethernet-1d.

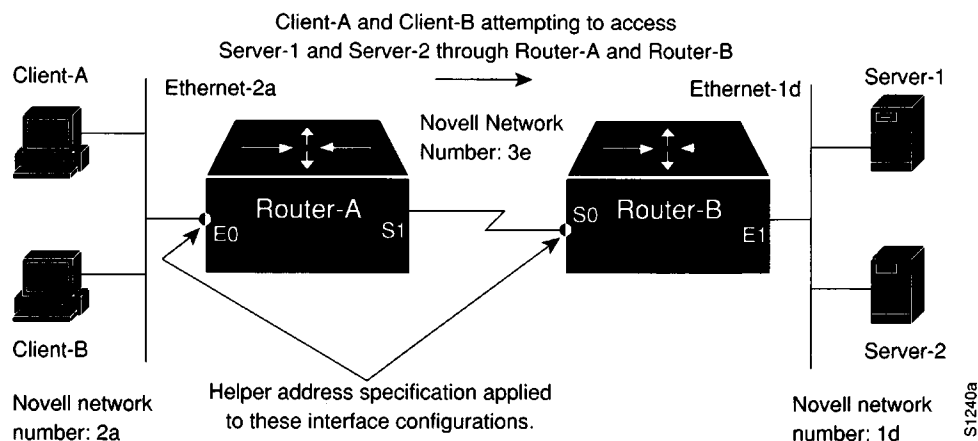


Figure 5-2 Single Serial Interconnection Helper Address Network

Assuming the use of the all nets broadcast address, the helper address specifications for the two routers would be as follows:

```
!Router-A helper address specification:
!
interface ethernet 0
 novell network 2a
 novell helper-address -1.ffff.ffff.ffff
!
!Router-B helper address specification:
!
interface serial 0
 novell network 3e
 novell helper-address -1.ffff.ffff.ffff
```

As in the prior example, -1 is used to specify flooding to all nets. Note that the helper address specification is required on Router-B because of the use of the all nets (-1) network specification (on Router-A). You can specify a specific network number as an alternative. In this case, you would specify Novell network 1d as follows on Router-A only:

```
!Router-A helper address specification:
!
interface ethernet 0
 novell network 2a
 novell helper-address 1d.ffff.ffff.ffff
```


Helper Address Configuration over Multiple Serial Interconnections

The key difference between the following example and prior examples is that Server-1 and Server-2 are now on separate Ethernet segments and are accessed by clients via separate routers (Router-B and Router-C.). Refer to Figure 5-3

There are two ways to propagate broadcasts: by using the all nets broadcast option or by using multiple, directed-broadcast helper address commands. If you use the all nets specification, all three routers must include helper address specifications.

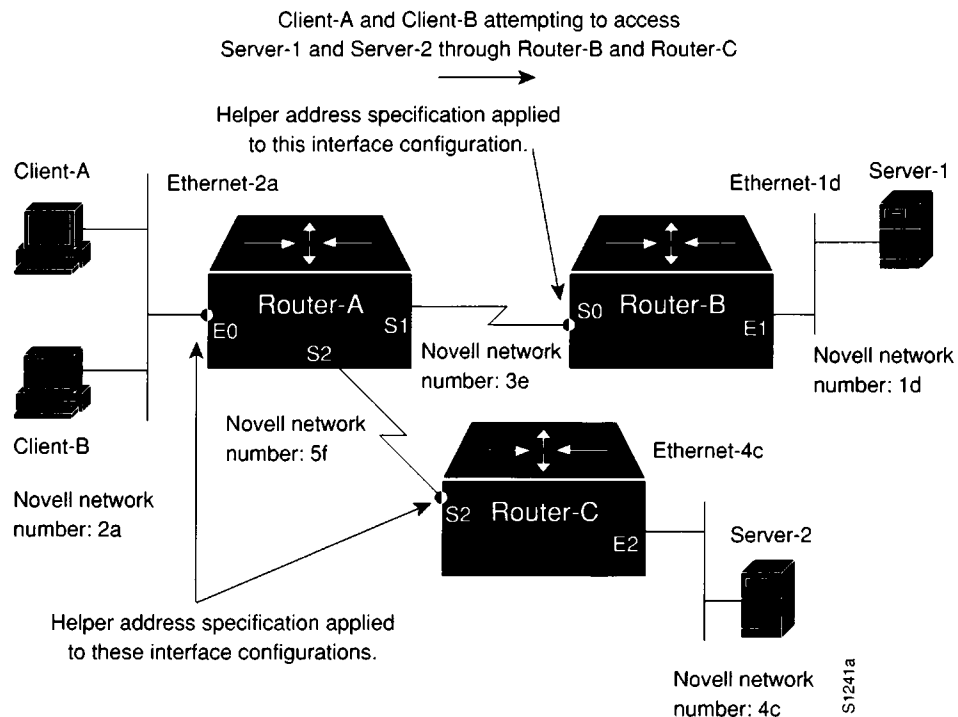


Figure 5-3 All Nets Multiple Serial Line Helper Address Specification

The all nets broadcast configurations for the router would be as follows:

```
!Router-A helper address specification:
!
interface ethernet 0
novell network 2a
novell helper-address -1.ffff.ffff.ffff
!
!Router-B helper address specification:
!
interface serial 0
novell network 3e
novell helper-address -1.ffff.ffff.ffff
!
!Router-C helper address specification:
!
interface serial 2
novell network 5f
novell helper-address -1.ffff.ffff.ffff
```

Note that the helper address specification is *required* on both Router-B and Router-C because of the use of the all nets (-1) network specification. on Router-A.

With Software Release 9.1, you can direct broadcasts to more than one network using multiple network number specifications on the same interface. Figure 5-4 illustrates an example where this multiple helper address command specification is applied.

Note: Prior to Software Release 9.1, Cisco routers only supported the assignment of a *single* helper address per interface. To allow clients to access servers on multiple segments required specifying the all nets broadcast (ffffff or -1) in a helper address. However, as of Software Release 9.1, multiple, directed broadcasts can be specified (on a per interface basis).

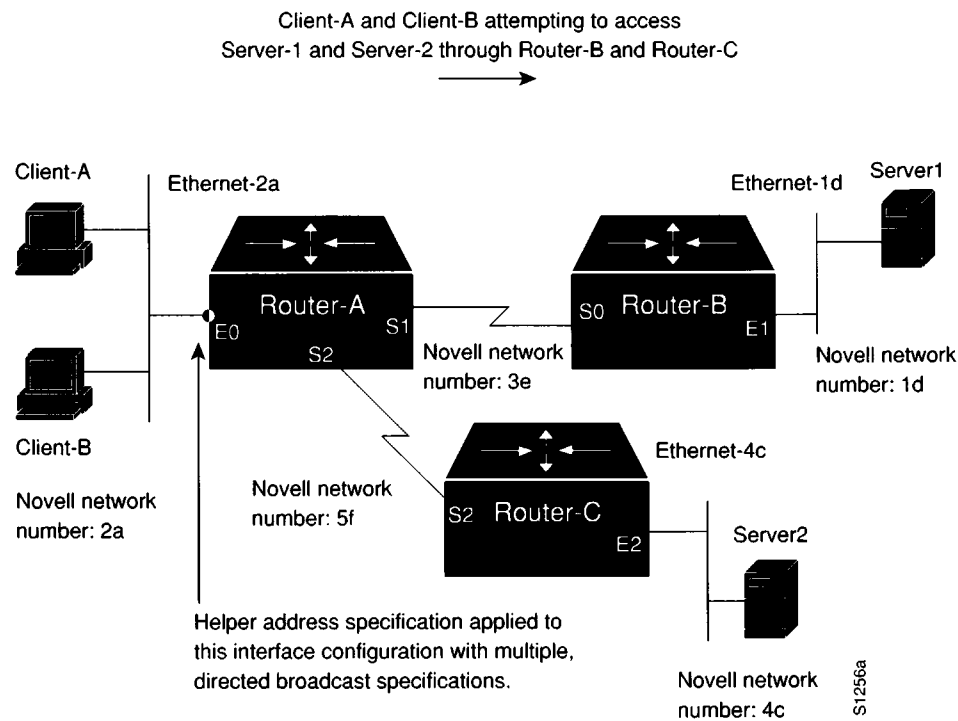


Figure 5-4 Directed Broadcast Helper Address Specification

The corresponding multiple helper address configuration would be modified as outlined in the following listing.

```
!Router-A helper address specification:
!
interface ethernet 0
novell network 2a
novell helper-address 1d.ffff.ffff.ffff
novell helper-address 4c.ffff.ffff.ffff
!
!Router-B helper address specification:
!
interface serial 0
novell network 3e
!
!Router-C helper address specification:
!
interface serial 2
novell network 5f
```

Helper Address Configuration for Reverse Broadcast NetBIOS Servers

In some cases, a NetBIOS server returns a broadcast response to client queries. This *reverse broadcast* must be handled similarly to normal helper address specifications for client queries.

Figure 5-5 illustrates a situation in which three sets of clients on three separate segments all must access a common file server (Server-X) on Ethernet-3F. In this case, the server supports a version of Novell NetBIOS in which the server broadcasts its response to the client queries. In this situation, Router-Hub must be configured with helper addresses on all four Ethernet interfaces illustrated in Figure 5-5.

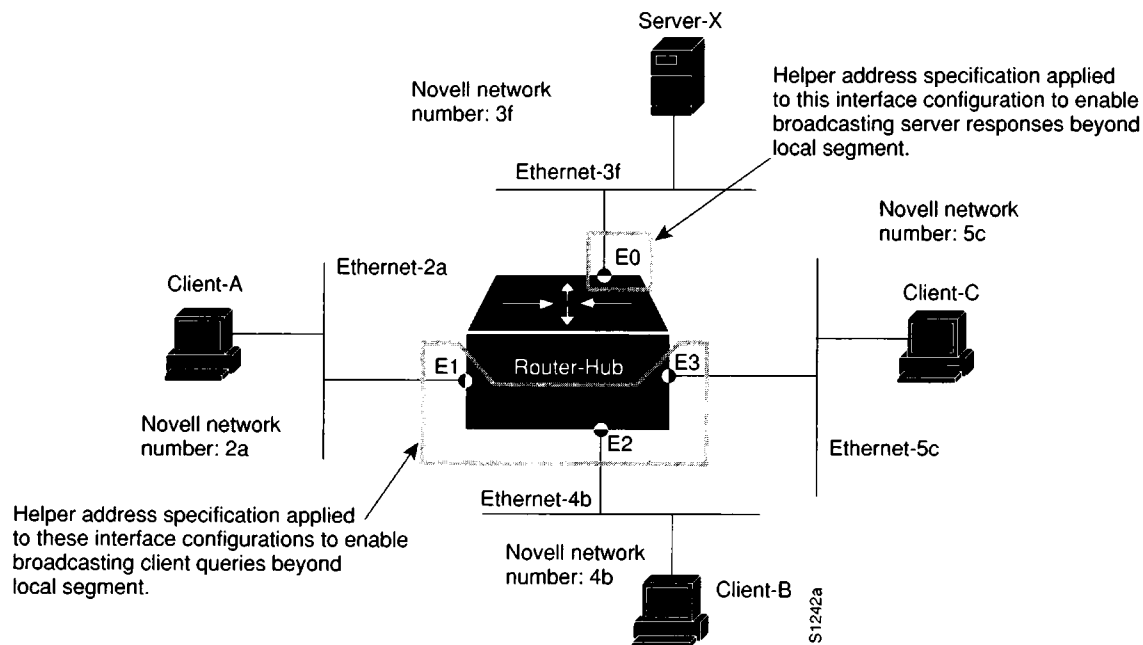


Figure 5-5 Reverse Broadcast Helper Address Network

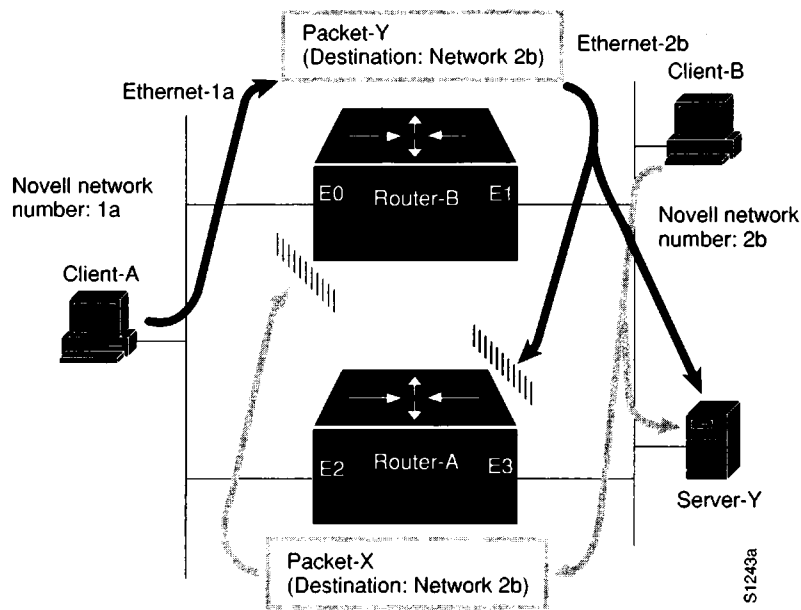
The Novell helper address configuration for the Router-Hub interfaces would be as follows:

```
!Router-Hub helper address specifications:
!
interface ethernet 0
novell network 3f
novell helper-address -1.ffff.ffff.ffff
!Required for reverse broadcast response from Server-X
!
interface ethernet 1
novell network 2a
novell helper-address -1.ffff.ffff.ffff
!Normal client broadcast specification
!
interface ethernet 2
novell network 4b
novell helper-address -1.ffff.ffff.ffff
!Normal client broadcast specification
!
interface ethernet 3
novell network 5c
novell helper-address -1.ffff.ffff.ffff
!Normal client broadcast specification
```

Note: Not all Novell NetBIOS servers require configuration of a helper address on the server interface. Consult with your host documentation to determine whether it generates a broadcast response (reverse broadcast) to client queries.

Helper Behavior with Parallel Routers

Use care in assigning broadcast-type helper addresses when Novell networks are interconnected over multiple routers and when file servers are using Novell NetBIOS. Although traffic will not permanently loop, local client queries can leak out through a router, resulting in a certain amount of excess traffic. Consider the situation illustrated in Figure 5-6.



Helper address specifications applied to all ethernet interfaces using all nets flooding option (1.ffff.ffff.ffff).

Figure 5-6 Novell Helper Address Handling with Parallel Routers

In this example, helper addresses are assigned to the Ethernet interfaces on Router-A and Router-B. The interface configurations might be as follows:

```
!Router-B helper address specifications:
!
interface ethernet 0
novell network 1a
novell helper-address -1.ffff.ffff.ffff
!
interface ethernet 1
novell network 2b
novell helper-address -1.ffff.ffff.ffff
!
!Router-A helper address specifications:
!
interface ethernet 2
novell network 1a
novell helper-address -1.ffff.ffff.ffff
!
interface ethernet 3
novell network 2b
novell helper-address -1.ffff.ffff.ffff
```

Consider what happens to Packet-Y from Client-A that is destined for Server-Y on Novell network 2b. Assuming that no access lists are in place, say that Router-B is the first to get a query from Client-A. Since the query is intended for an offnet host, Router-B broadcasts the query out Ethernet interface E1 and onto Novell network 2b. The broadcast finds its way to Server-Y (hopefully causing a response, assuming Server-Y is operational) and also lands at Ethernet interface E3 on Router-A. There the packet is dropped. This is the expected behavior.

Note: Server-Y actually receives two copies of Packet-Y, one via Router-A and one via Router-B. The server's response depends on its application implementation.

Now consider Packet-X, a client query from Client-B also intended for Server-Y. In this case, the broadcast packet finds its server on the same wire to which it is connected. However, Router-A forwards this broadcast, because the source address is local—which puts the locally targeted packet onto Novell network 1a. This packet will continue to propagate outward through the network until the internet terminates (or until the packet has traversed 15 routers), but it will not leak back into Novell network 2b because the routers see that the source network in the packet is 2b. In no case is the packet sent back along the path to the source network of the packet. In the preceding example, the packet would be dropped when it reaches Ethernet interface E0 on Router-B.

This situation is a type of *partial* loop. True routing loops are prevented, but some excess traffic is created.

Note: To prevent this from happening, you can use network-specific broadcasts. However, you may not be able to do so if many clients and servers must access each other on segments separated by parallel routers. Depending on your configuration, increasing the number of paths the router can store for each destination may reduce the amount of excess traffic. Use the **novell maximum-paths** command to do this.

Clients Cannot Connect to Server over Packet-Switched Network

Symptom: Local servers are responding, but servers on the other side of a packet-switching network that interconnects routers do not respond. Router *appears* to block IPX over the packet-switched network.

Possible Causes and Suggested Actions

Table 5-4 outlines possible causes of blocked access to a Novell servers over packet-switched networks.

Table 5-4 Causes and Actions for Blocked Novell Traffic over PSNs

Possible Cause	Suggested Actions
X.25 address mapping error	<p>Step 1: Using write terminal command, examine configuration of router.</p> <p>Step 2: Make sure that the MAC addresses and X.121 addresses specified in any x25 map novell interface subcommands match the addresses associated with the respective destination routers.</p> <p>(Refer to “Notes About PSN Address Map Specifications” immediately following this table for illustrative discussions regarding address mapping).</p>
Frame Relay address mapping error	<p>Step 1: Using write terminal command, examine configuration of router.</p> <p>Step 2: Make sure that the MAC addresses and DLCI addresses specified in any frame-relay map novell interface subcommands match the addresses associated with the respective destination routers.</p> <p>(Refer to “Notes About PSN Address Map Specifications” immediately following this table for illustrative discussions regarding address mapping).</p>
Misconfigured network number specification on servers or on routers	<p>Step 1: Refer to Table 5-1 and the symptom discussion for “Clients Cannot Communicate with NetWare Servers over Router” for a discussion about diagnosing and resolving this problem.</p>
Encapsulation error	<p>Step 1: Use write terminal or show interfaces command to determine encapsulation type.</p> <p>Step 2: Look for relevant packet-switching encapsulation type (such as x25 encapsulation or frame-relay encapsulation).</p> <p>Step 3: If an encapsulation command is not present, the default is HDLC encapsulation.</p> <p>Step 4: For PSN interconnection, you must explicitly specify an encapsulation type.</p>

Notes About PSN Address Map Specifications

When routing Novell IPX (or any protocol) over a packet-switched network, you must specify mapping between Novell and packet-switched network addresses. Consider the two examples illustrated in Figure 5-7 and Figure 5-8. Figure 5-7 illustrates an address map specification when routing Novell IPX over an X.25 PSN, while Figure 5-8 illustrates an address map specification when routing Novell IPX over a Frame Relay network. Relevant configurations and a brief explanation of command variables are provided in the following discussions. Refer to the *Router Products Configuration and Reference* publication for more information about address map specifications.

Address Mapping for Novell-to-X.25 Interconnection

As illustrated in Figure 5-7, Novell-to-X.25 address map specifications would be required for both Router-A and Router-B.

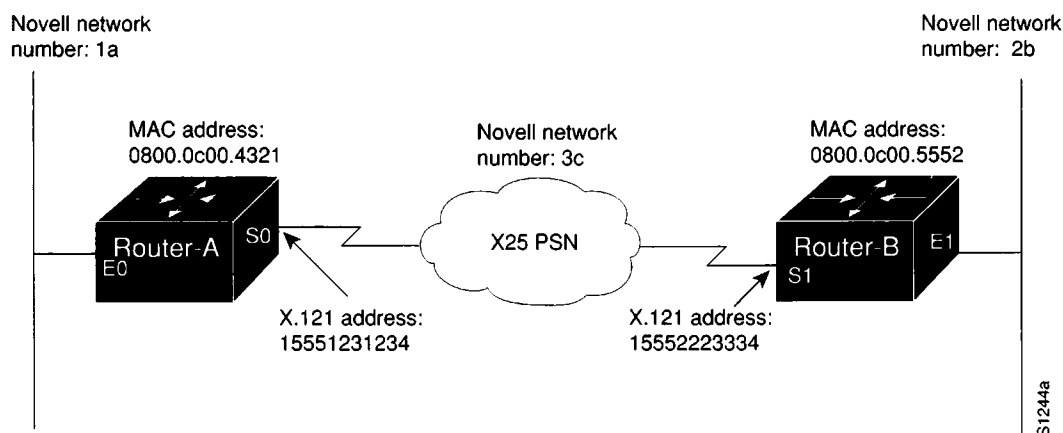


Figure 5-7 Example Network Diagram Illustrating Novell-to-X.25 Mapping

The specific interface specifications would be as follows:

```
!  
interface serial 0  
x25 map novell 3c.0200.0c00.5552 15552223334 broadcast  
! Above specifies Novell-to-X.121 address map configuration for Router-A  
  
interface serial 1  
x25 map novell 3c.0800.0c00.4321 15551231234 broadcast  
! Above specifies Novell-to-X.121 address map configuration for Router-B
```

In the preceding configurations, the MAC address is obtained using the **write terminal** command on the target router. Look for the **novell routing** command in the configuration listing. It is displayed with the auto-generated MAC address appended to the command. For example, for Router-A in Figure 5-7, you would see the following listing:

```
novell routing 0800.0c00.4321
```


Address Mapping for Novell-to-Frame Relay Interconnection

Figure 5-8 illustrates essentially the same interconnection arrangement as illustrated for the preceding X.25 example address mapping configuration, except that the PSN used is a Frame Relay network. In an analogous manner, Novell-to-Frame Relay address map specifications would be required for both Router-A and Router-B.

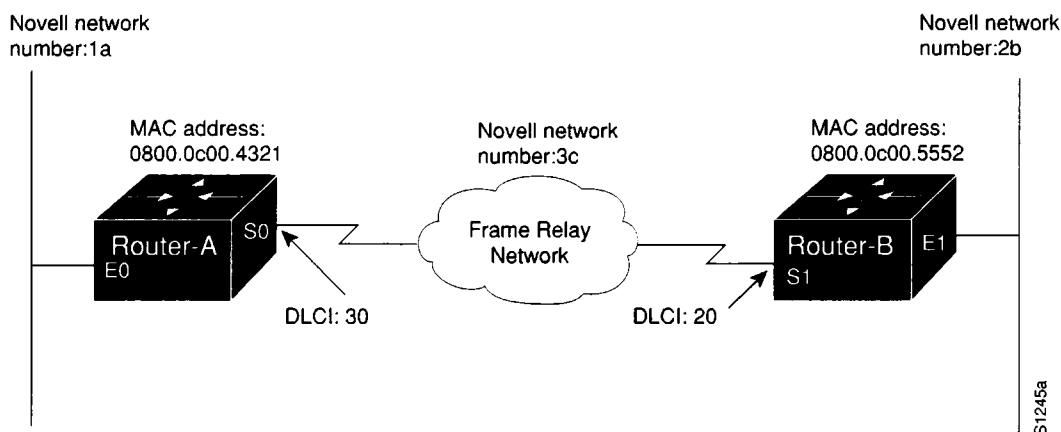


Figure 5-8 Example Network Diagram Illustrating Novell-to-Frame Relay Mapping

The specific interface configurations would be as follows:

```
!
interface serial 0
frame-relay map novell 3c.0800.0c00.5552 20 broadcast
! Above specifies Novell-to-DLCI address map configuration for Router-A

interface serial 1
frame-relay map novell 3c.0800.0c00.4321 30 broadcast
! Above specifies Novell-to-DLCI address map configuration for Router-B
```

In these example configurations, the MAC address is obtained in the same manner as described in the X.25 example: use the **write terminal** command on the target router and look for the **novell routing** command in the configuration listing.

Chapter 6

Troubleshooting TCP/IP Connectivity

6

TCP/IP Internet Diagnostic Overview 6-1

Problem Isolation in TCP/IP Networks 6-2

TCP/IP Connectivity Symptoms 6-3

Symptom Summary 6-3

Host Cannot Access Offnet Host(s) 6-4

Possible Causes and Suggested Actions 6-4

Host Cannot Access Certain Networks 6-6

Possible Causes and Suggested Actions 6-6

Connectivity Available to Some Hosts, but Not Others 6-7

Possible Causes and Suggested Actions 6-7

Some Services Are Available, Others Are Not 6-8

Possible Causes and Suggested Actions 6-8

Users Cannot Make Connections When One Path is Down 6-9

Possible Causes and Suggested Actions 6-10

Router Sees Duplicate Routing Updates and Packets 6-11

Possible Causes and Suggested Actions 6-11

Routing Works for Some Protocols, Not for Others 6-12

Possible Causes and Suggested Actions 6-12

Router/Host Cannot Reach Certain Parts of Its Own Network 6-13

Possible Causes and Suggested Actions 6-13

Note About IP Addresses and Subnet Masks 6-14

Traffic Is Not Getting Through Router Using Redistribution 6-15

Possible Causes and Suggested Actions 6-15

Chapter 6

Troubleshooting TCP/IP Connectivity

6

This chapter presents protocol-related troubleshooting information for TCP/IP connectivity problems. The emphasis here is on symptoms and problems associated with TCP/IP network connectivity.

This chapter consists of the following sections:

- Diagnostic tools and techniques used for isolating TCP/IP-related problems
- Overview of symptom list
- Symptom/problem/action modules

The problem/solution modules consist of the following sections:

- Symptom statement—A specific symptom associated with the technology/media/protocol in which this module appears.
- Possible causes and suggested actions—A table for each symptom containing possible causes for the symptom and suggested actions for resolving each cause.

TCP/IP Internet Diagnostic Overview

Some of the world's largest networks today rely on the TCP/IP suite of networking protocols. With a relatively small kit of basic tools, network managers can learn a lot about what is going on (or wrong) in an internetwork. The following tools (all available via the router's basic management interface) form the core of the administrator's internetwork toolkit:

- **ping** and **trace** commands—The **ping** and **trace** commands are essential in determining where failures are occurring.
- **show** commands—The **show** commands provide information about interface conditions, protocol status, neighbor reachability, router configuration and status, level of traffic, errors and drops, and other network data.
- **debug** commands—The **debug** commands provide clues about the status of both the router and other network entities. Use **debug** commands with great care. Using the wrong **debug** at the wrong time can exacerbate already poorly performing networks.



Caution: Throughout this and other chapters, the use of **debug** commands is suggested for obtaining information about network traffic and router status. Use these commands with great care. In general, it is recommended that these commands only be used under the direction of your router technical support representative when troubleshooting specific problems. Enabling debugging can disrupt operation of the router, when internets are experiencing high load conditions. When you finish using a **debug** command, remember to disable it with its specific **undebug** command or with the **undebug all** command.

When specific diagnostic commands are considered particularly useful for troubleshooting symptoms, they are listed with the specific symptom discussion in this chapter.

Refer to “Using Cisco Diagnostic Tools” in Chapter 1 of this publication for general information about using these tools. The **debug** commands discussed in this publication are described in Chapter 10, “Debug Command Reference,” while the remainder of the diagnostic commands are detailed in *Router Products Configuration and Reference* publication.

Problem Isolation in TCP/IP Networks

One important consideration to remember when troubleshooting broken interconnections is that normally everything does not break at the same time. As a result, when trying to isolate a problem, you can typically work out from an operational node to the point of failure. The following basic steps should help when trying to isolate the source of connection disruption:

- Step 1:** First, determine whether your local host is properly configured (for instance, correct subnet mask configurations and correct default gateway specifications).
- Step 2:** Next, use the **ping** or **trace EXEC** commands to determine whether the routers through which you must communicate can respond. Start with the most local router and progressively “ping out” through the internet.
- Step 3:** If you cannot get through a particular node, examine the node’s configuration and use the various **show** commands to determine the router’s state.
- Step 4:** If you can get to all the routers in the path, check the host configuration at the remote host (or get someone’s help to do so) and check its configuration.
- Step 5:** Check the routers’ routing tables (**show ip route**) and ARP tables (**show ip arp**) for any anomalies (such as duplicate routes) and to see if the host(s) in question are appearing. Another useful diagnostic command is **show ip cache**, which shows the routing table cache used to fast-switch IP traffic.

TCP/IP Connectivity Symptoms

The symptom modules that follow pertain to TCP/IP internetwork problems. Unless otherwise indicated, each module is presented as a set of general problems. Where there are special considerations associated with a situation, notes are included.

Symptom Summary

The following TCP/IP connectivity symptoms are discussed in this section:

- Host cannot access offnet host(s) through router.
- Host cannot access certain networks via router.
- Users can access some hosts, but not others.
- Some services are available, others are not.
- Users cannot make any connections when one parallel path is down.
- Router sees duplicate routing updates and packets.
- Certain protocols are being routed, others are not.
- Router or other host cannot reach certain parts of its own network.
- Routing not working when redistribution is used.

Note: The symptoms that follow are generic in nature. However, when host configuration problems are discussed, they are addressed assuming UNIX end systems. Equivalent kinds of actions may be applicable to non-UNIX hosts as well, but the discussion here does not address non-UNIX end station problems.

Host Cannot Access Offnet Host(s)

Symptom: Host-A is unable to communicate with Host-B on another network. Here, when you attempt to make a connection to some intervening router, you may or may not be able to make a successful connection. In either case, you are unable to connect to the target host on the other side of the router. This situation is illustrated in Figure 6-1.

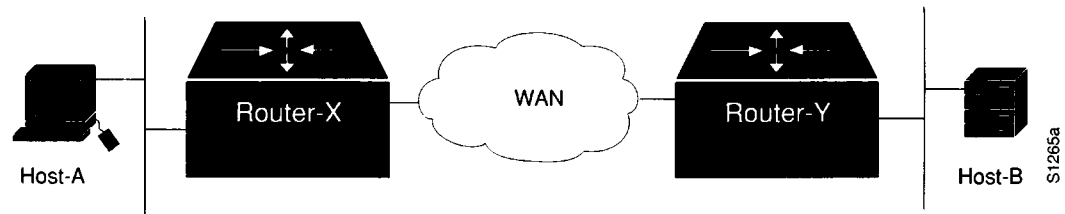


Figure 6-1 Host-A Cannot Communicate with Host-B over Routers

Possible Causes and Suggested Actions

Table 6-1 outlines possible causes of blocked access to a specific host on a remote network.

Table 6-1 Causes and Actions for Blocked Access to Remote Hosts

Possible Cause	Suggested Actions
No default gateway specification	<p>Step 1: Determine whether a default gateway is included in the routing table of the host attempting to make a connection (Host-A in Figure 6-1). Use the following UNIX command:</p> <pre>netstat -rn</pre> <p>Step 2: Inspect the output of this command for a default gateway specification.</p> <p>Step 3: If the specified default gateway is incorrect, or if it is not present at all, you can change or add a default gateway using the following UNIX command at the local host:</p> <pre>route add default address 1</pre> <p>(<i>address</i> is the IP address of the default gateway; the value 1 indicates that the specified node is one hop away)</p> <p>Step 4: To automate this as part of the boot process, specify the default gateway's IP address in the following UNIX host file:</p> <pre>/etc/defaultrouter</pre>
Misconfigured subnet mask	<p>Step 1: Check the following two locations for possible subnet mask errors:</p> <ul style="list-style-type: none">■ <code>/etc/netmasks</code>■ <code>/etc/rc.local</code> <p>Step 2: Fix if incorrectly specified or add if not included.</p>

Possible Cause	Suggested Actions
Router between hosts is down	<p><i>Step 1:</i> Use the ping command to determine whether the router is reachable.</p> <p><i>Step 2:</i> If the router does not respond, isolate the problem and repair the broken interconnection.</p> <p><i>Step 3:</i> Refer to the brief discussion at the beginning of this chapter entitled “Problem Isolation in TCP/IP Networks” and to discussions in Chapter 1 for more information.</p>

Host Cannot Access Certain Networks

Symptom: Host cannot access certain other networks on the other side of a router. Some networks might be accessible.

Possible Causes and Suggested Actions

Table 6-2 outlines possible causes of unreachable network problems.

Table 6-2 Causes and Actions for Unreachable Network Problems

Possible Cause	Suggested Actions
No default gateway	<p>Step 1: Check the host for proper default gateway specification as described in the preceding symptom section “Host Cannot Access Offnet Host(s).”</p> <p>Step 2: Modify or add default gateway specification as required. Table 6-1 provides more details regarding default gateway specification.</p>
Bad access list (getting routing information for some routes, but not others)	<p>Step 1: Check routing table with show ip route command and protocol exchanges with the appropriate debug command (such as debug ip-igrp and debug ip-rip).</p> <p>Step 2: Look for information concerning the specific network with which you are unable to communicate.</p> <p>Step 3: Check the use of access lists on the routers in the path and make sure that a distribute-list or distance command does not filter out the route.</p> <p>Step 4: Temporarily disable access lists (by removing ip access-group commands) and use the trace or ping command with record route option set to determine whether traffic can get through when the access list is removed.</p>
Discontinuous network addressing due to poor design	<p>Step 1: Use the show ip route command to examine which routes are known and how they are being learned.</p> <p>Step 2: Use the trace or ping commands to see where traffic is stopping.</p> <p>Step 3: Fix topology or reassign addresses to include all appropriate network segments into the same major network. Refer to the symptom section “Users Cannot Make Connections When One Path is Down” later in this chapter for additional information.</p>
Discontinuous network addressing due to link failure	<p>Step 1: Restore disabled link.</p> <p>Step 2: If the loss of a link occurs and you cannot use a parallel path, examine network address assignments.</p> <p>Step 3: If link failure results in a discontinuous network because one network has different points of contact with two now isolated subnets of a different major net, assign secondary addresses along the backup path to restore major network connectivity.</p>

Connectivity Available to Some Hosts, but Not Others

Symptom: Hosts on a network can communicate with specific hosts on the other side of a router, but are unable to communicate with certain other hosts.

Possible Causes and Suggested Actions

Table 6-3 outlines possible causes of selectively blocked access to hosts.

Table 6-3 Causes and Actions for Selectively Blocked Host Access

Possible Cause	Suggested Actions
Misconfigured subnet mask	<p>Step 1: Check subnet masks on hosts and routers.</p> <p>Step 2: Look for mismatch between subnet mask. What may be a specific host address to one host may turn into a subnet broadcast when a different mask is applied at a router.</p> <p>Step 3: Fix subnet mask on host or router as required.</p>
Bad access list (host is denied by some router in the path)	<p>Step 1: Ping out through path to determine where packets are being dropped.</p> <p>Step 2: If you can identify the router that is stopping traffic, use the write terminal EXEC command to see whether an access list is being used. You can also use the show access-lists and show ip interface commands in combination to determine whether access lists are being used.</p> <p>Step 3: Temporarily disable the access list.</p> <p>Step 4: See whether traffic can get through the router (ping or Telnet).</p> <p>Step 5: If traffic can get through, carefully review the access list and its associated commands for proper authorization.</p>
Missing default gateway specification on remote host	<p>Step 1: Determine whether you can access any offnet hosts from inaccessible remote host (may need system admin at remote end to log in to inaccessible hosts).</p> <p>Step 2: Check the remote host for proper default gateway specification as described in the earlier symptom section "Host Cannot Access Offnet Host(s)."</p> <p>Step 3: Modify or add default gateway specification as required.</p>

Some Services Are Available, Others Are Not

Symptom: In some cases you may be able to get through to hosts using some protocols, but cannot get through using others. For instance, you may be able to ping a host and FTP to a host, but Telnet does not get through.

Possible Causes and Suggested Actions

Table 6-4 outlines possible causes of some host services being functional, while others are not.

Table 6-4 Causes and Actions for Selective Service Availability

Possible Cause	Suggested Actions
Misconfigured extended access list	<p>Step 1: Use the trace command to determine path taken to reach remote hosts.</p> <p>Step 2: (Optional) On each router in the path, enable debug ip-icmp command.</p> <p>Any router that returns “unreachables” is suspect.</p> <p>Step 3: If you can identify the router that is stopping traffic, use the write terminal EXEC command to see whether an access list is being used. You can also use the show access-lists and show ip interface commands in combination to determine whether access lists are being used.</p> <p>Step 4: Temporarily disable the access list.</p> <p>Step 5: See whether traffic can get through the router.</p> <p>Step 6: If traffic can get through, carefully review the access list and its associated commands for proper authorization.</p> <p>Step 7: In particular, look for TCP port configured in extended access lists.</p> <p>Step 8: If ports are specified, be sure that all needed ports are explicitly permitted by access lists.</p>

Users Cannot Make Connections When One Path is Down

Symptom: In configurations featuring multiple paths between networks, when one of the parallel links breaks, there is no communication via the alternative routes.

Figure 6-2 illustrates one example of a situation in which this can occur. Here, one major network (Net-B) has two or more access points into another major network (Net-C), while a third link joins two separate subnets of Net-C. Details are provided in the “Possible Causes and Suggested Actions” discussion that follows.

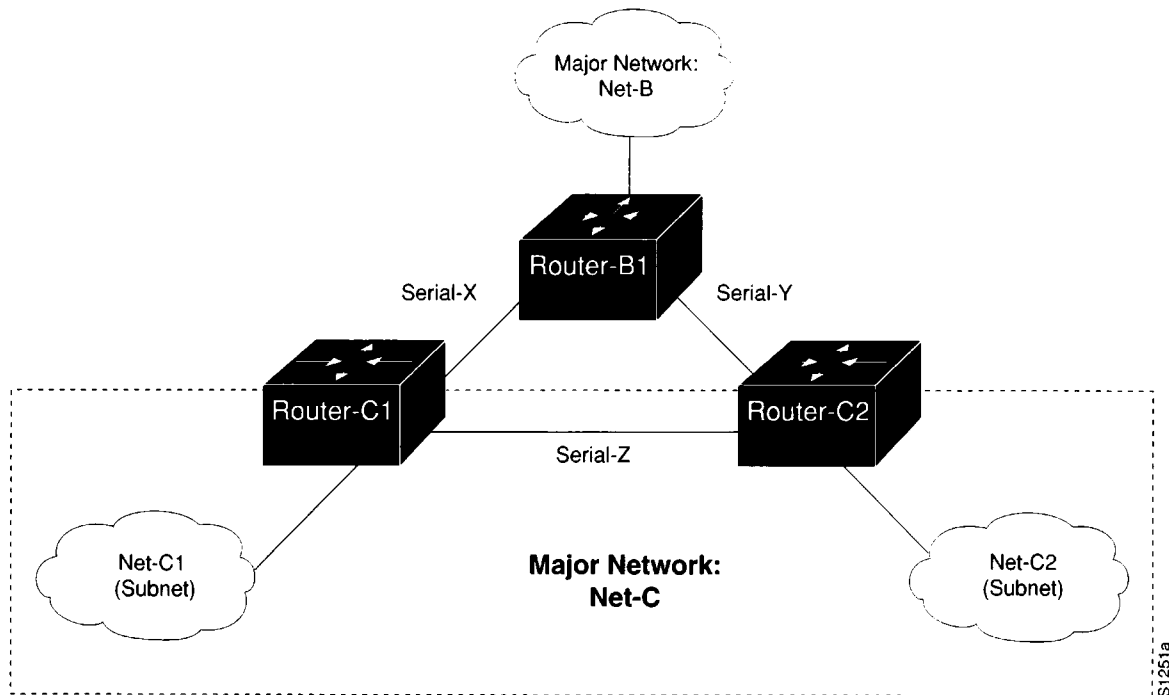


Figure 6-2 Problem Parallel Path Configuration Example

Possible Causes and Suggested Actions

Table 6-5 outlines possible causes of completely blocked connectivity when *only* one parallel link is lost.

Table 6-5 Causes and Actions for an Inadvertently Blocked Parallel Path

Possible Cause	Suggested Actions
Discontinuous network due to failure. If Serial-Z is lost, traffic cannot traverse from Net-C1 to Net-C2 through Router-B1	<p>Step 1: Bring link back up.</p> <p>Step 2: As an alternative, ensure that all interfaces are included in the same major network using a secondary IP address configuration. Refer to Figure 6-2. If Serial-Z is lost, Major Network Net-C becomes a discontinuous network because Router-B1 is separating the two Net-C subnets (Net-C1 and Net-C2). Traffic between Router-C1 and Router-C2 will not get through Router-B1 because Router-B1 assumes they are directly connected.</p>
Routing has not converged	<p>Step 1: Assuming you have used secondary addresses, examine routing tables for routes that are listed as “possibly down.” If this entry is found, routing protocol has not converged.</p> <p>Step 2: Wait for the routing protocol to converge. Examine the routing table later.</p>
Misconfigured access lists or other routing filters	<p>Step 1: Check for access lists in the secondary path.</p> <p>Step 2: If present, disable and determine whether traffic is getting through.</p> <p>Step 3: If traffic is getting through, this suggests access list and accompanying commands are causing traffic stoppage.</p> <p>Step 4: Evaluate and fix access lists as necessary.</p>
Errors on serial link	<p>Step 1: If the link is a serial link, look for input on the interface (using the show interfaces serial number command).</p> <p>Step 2: Refer to the discussions regarding serial debugging in Chapter 7, “Troubleshooting WAN Connectivity,” and Chapter 1, “Troubleshooting Overview,” for more information.</p>
Errors on Ethernet link	<p>Step 1: Use a TDR to find any unterminated Ethernet cable.</p> <p>Step 2: Check host cables and transceiver cables to determine whether any are incorrectly terminated, overly long, or damaged.</p> <p>Step 3: Look for a jabbering transceiver attached to a host (may require host-by-host inspection).</p>

Router Sees Duplicate Routing Updates and Packets

Symptom: When the router sees duplicate routing updates, your network users are also apt to experience sudden loss of connections and extremely poor performance. Here, the router sees other routers and hosts on multiple interfaces.

Possible Causes and Suggested Actions

Table 6-6 outlines possible causes of routers seeing duplicate routing updates and packets.

Table 6-6 Causes and Actions for Duplicate Routing Updates and Packets

Possible Cause	Suggested Actions
Bridge or repeater in parallel with router, causing updates and traffic to be seen as coming from both sides of an interface	Step 1: To determine whether this is the problem, use the show ip protocol EXEC command to get a list of routers from which the router is obtaining route information.
	Step 2: Look for routers that are known to be remote to the network connected to the router.
	Step 3: If you see routers listed but know them not to be attached to any directly connected networks, this is a likely problem.
	Step 4: Another test is to use the show ip route command to examine routes for each interface.
	Step 5: Look for paths to the same networks with the same cost on multiple interfaces.
	Step 6: If you determine that there is a bridge in parallel, remove the bridge or configure access filters (on the bridge) that block routing updates.

Routing Works for Some Protocols, Not for Others

Symptom: For instance, Telnet works from a host on one network to a host on another network on the other side of a router. Perhaps Domain Name Service (DNS) works with your own domain, but does not work for external domains.

Possible Causes and Suggested Actions

Table 6-7 outlines possible causes for some protocols not working over a TCP/IP internetwork, when other protocols are working.

Table 6-7 Causes and Actions for Some Protocols Not Being Routed

Possible Cause	Suggested Actions
Misconfigured access list	<p>Step 1: Use ping and trace EXEC commands as described at the beginning of this chapter to isolate the router with the misconfigured access list.</p> <p>Step 2: Once you determine where traffic is stopping, review the configuration of that router using the write terminal EXEC command.</p> <p>Step 3: Look for any access list in the configuration.</p> <p>Step 4: Temporarily disable the access list and monitor traffic to and through the suspect router.</p> <p>Step 5: If the router is allowing previously blocked traffic through, some kind of error in the access list is probably at fault.</p> <p>Step 6: Make sure you explicitly permit desired traffic, or that traffic will be blocked with the implicit deny statement ending all access lists.</p>

Router/Host Cannot Reach Certain Parts of Its Own Network

Symptom: A router or host is unable to communicate with other routers or hosts known to be directly connected to the same network.

Possible Causes and Suggested Actions

Table 6-8 outlines possible causes of hosts/routers being unable to reach routers/hosts in the same major network.

Table 6-8 Causes and Actions for Unreachable Hosts on Same Major Network

Possible Cause	Suggested Actions
Subnet mask configuration mismatch between router and host	<p>Step 1: Ping out to destination from your host/router, as discussed in “TCP/IP Internet Diagnostic Overview” at the beginning of this chapter.</p> <p>Step 2: If you can ping from the local host to the local router (but not to the remote host), and you can ping from the local router to the remote host, there is probably a subnet mask configuration problem on your local host or router.</p> <p>Step 3: Check host and router configurations for subnet mask mismatch. Make sure that all subnet masks match.</p> <p>(Note that masks may not match if proxy ARP is being used. Refer to RFC 1027 for more information about using proxy ARP.)</p> <p>Refer to “Host Cannot Access Offnet Host(s)” for notes about host subnet mask.</p> <p>Refer to “Note About IP Addresses and Subnet Masks” immediately following this table for comments about subnet mask conflicts.</p>
Misconfigured access list	<p>Step 1: Use ping and trace EXEC commands as described at the beginning of this chapter to isolate the router with the misconfigured access list.</p> <p>Step 2: Once you determine where traffic is stopping, review the configuration of that router using the write terminal EXEC command.</p> <p>Step 3: Look for any access lists in the configuration.</p> <p>Step 4: Temporarily disable the access list and monitor traffic to and through the suspect router.</p> <p>Step 5: If the router is allowing previously blocked traffic through, some kind of error in the access list is probably at fault.</p>
No default gateway specified	<p>Step 1: Check the remote host for proper default gateway specification as described in the earlier symptom section “Host Cannot Access Offnet Host(s).”</p>

Possible Cause	Suggested Actions
	<p>Step 2: Check configuration on hosts and routers for static routes.</p> <p>If static routes exist and no default gateway is specified, access to some hosts and routers might be possible, while others are unavailable. To resolve this inconsistency, you have several options:</p> <ul style="list-style-type: none"> ■ Specify a default gateway on your host as described in the symptom section “Host Cannot Access Offnet Host(s)” earlier in this chapter. ■ Enable proxy ARP on the host; make the local cable the default network (network 0 for RIP). ■ Run the Gateway Discovery Protocol (GDP) on the host (BSD UNIX host only). Allows dynamically defined default gateway. ■ Run a routing protocol (such as RIP) on the host. (Note that there is high host processing overhead associated with this option.)

Note About IP Addresses and Subnet Masks

In most IP networks, routers and hosts should agree on their common subnet mask. If a router and a host disagree on the length of the subnet mask, then packets may not be routed correctly. Consider the situation in Table 6-9.

A host interprets a particular address (192.31.7.49) as being Host 1 on the third subnet (subnet address 48). However, the router interprets this same address as belonging to Host 17 on the first subnet (subnet address 32). The result is that any packet destined for 192.31.7.49 will either be sent out an incorrect interface or dropped, depending on the configuration of the router.

Table 6-9 Comparison of Host and Router Subnet Mask Effects

Routing Info	Host Value	Router Value
Destination IP Address	192.31.7.49	192.31.7.49
Subnet Mask	255.255.255.240	255.255.255.224
Interpreted Address	Subnet address 48, host 1	Subnet address 32, host 17

Traffic Is Not Getting Through Router Using Redistribution

Symptom: Traffic is not getting through a router that is redistributing routes between two different routing domains—typically RIP and IGRP. Observed symptoms range from bad performance (if nonoptimal routes are being used because the best path is blocked by a misconfigured redistribution) to no communication at all.

Possible Causes and Suggested Actions

Table 6-10 outlines possible causes of routing problems stemming from route redistribution.

Table 6-10 Causes and Actions for Route Redistribution Problems

Possible Cause	Suggested Actions
Missing default-metric command	<p>Step 1: Check router configuration using write terminal EXEC command.</p> <p>Step 2: If default metric command is missing, add the appropriate version. Refer to the <i>Router Products Configuration and Reference</i> publication for details.</p>
Problem with the default administrative distance	<p>Step 1: Determine the policy for identifying how much you <i>trust</i> routes derived from different domains.</p> <p>Problems occur when a particular route is, by default, trusted less than another, but actually is the preferred route.</p> <p>Step 2: As applicable, use the distance router subcommand to vary the level of trust associated with specific routing information.</p> <p>Again, refer to the <i>Router Products Configuration and Reference</i> publication for details.</p>

Chapter 7

Troubleshooting WAN Connectivity

7

Diagnosing WAN and Serial Line Problems 7-1

Using the Show Interfaces Command to Troubleshoot Serial Lines 7-2

- Deciphering Serial Line Status Diagnostics 7-3

- Basic Serial Diagnostic Fields 7-6

 - Evaluating Input Errors 7-6

 - Evaluating Output Drops 7-8

 - Evaluating Input Drops 7-9

 - Evaluating Interface Resets 7-9

 - Evaluating Carrier Transitions 7-10

- WAN-Specific Diagnostic Fields 7-11

 - Example: X.25 Show Interfaces Diagnostic Fields 7-11

Using the Show Controllers Command to Troubleshoot Serial Lines 7-13

Using Debug Commands to Troubleshoot Serial Lines 7-13

Special Serial Line Tests 7-14

- CSU/DSU Local and Remote Loopback Tests 7-15

 - CSU/DSU Local Loopback Tests for HDLC Links 7-15

 - CSU/DSU Remote Loopback Tests for HDLC Links 7-16

- Extended Ping Tests 7-16

- Troubleshooting Clocking Problems 7-18

 - Clocking Problem Causes 7-18

 - Detecting Clocking Problems 7-18

 - Isolating Clocking Problems 7-19

 - Suggested Clocking Problem Remedies 7-19

- Adjusting Buffers to Ease Overutilized Serial Links 7-20

 - Managing System Buffers 7-20

 - Implementing Hold Queue Limits 7-22

 - Using Priority Queuing to Reduce Bottlenecks 7-23

WAN and Serial Line Connectivity Symptoms 7-25

Intermittent Connectivity 7-26

- Possible Causes and Suggested Actions 7-26

Connections Die as Load Increases 7-27

- Possible Causes and Suggested Actions 7-27

Connections Die at a Particular Time of Day 7-28

Possible Causes and Suggested Actions 7-28

Connections Die After Some Period of Normal Operation 7-29

Possible Causes and Suggested Actions 7-29

Users Cannot Connect to Resources over New HDLC Link 7-30

Possible Causes and Suggested Actions 7-30

Users Cannot Connect to Resources over New X.25 WAN Link 7-31

Possible Causes and Suggested Actions 7-31

Users Cannot Connect to Resources over New Frame Relay Link 7-33

Possible Causes and Suggested Actions 7-33

Users Cannot Connect to Resources over New SMDS Link 7-35

Possible Causes and Suggested Actions 7-35

Some Users Cannot Connect to Resources over WAN 7-37

Possible Causes and Suggested Actions 7-37

Chapter 7

Troubleshooting WAN Connectivity

7

This chapter presents basic WAN and serial line diagnostics, a series of common symptoms, possible causes of those symptoms, and general suggestions for identifying, isolating, and resolving those causes. The emphasis here is on symptoms and problems associated with WAN connectivity.

This chapter consists of the following sections:

- Diagnostic tools and techniques used for isolating WAN problems
- WAN connectivity symptom list
- Symptom/cause/action modules

The symptom/cause/action modules consist of the following sections:

- Symptom statement—A specific symptom associated with the technology/media/protocol in which this module appears.
- Possible causes/suggested actions—For each possible cause, a suggested action is specified, with brief suggestions regarding how you can determine whether the specific cause is causing connectivity problems and what to do to resolve it.

Diagnosing WAN and Serial Line Problems

When troubleshooting WAN problems, there are a variety of tools and techniques that can be applied to isolate the cause of interruptions. Some commonly used techniques are outlined in the discussion that follows. Where appropriate, pointers to other parts of this manual are included. In addition, when information specific to a particular technology, media, or protocol is needed, supporting notes are included in the text.

The following tools are universally applicable when gathering information to troubleshoot WAN links:

- **show interfaces serial** *number*, **show controllers mci**, and **debug** diagnostic EXEC commands. Refer to the *Router Products and Configuration Reference* publication for complete details about required variables and options for **show** commands. Details for **debug** commands are provided in Chapter 10 of this publication.
- CSU/DSU loopback tests

- Ping (Echo Request/Echo Reply) tests
- Serial line clock troubleshooting techniques

The application of these tools is described in the discussions that follow.

Note: The emphasis here is on describing Cisco-specific tools. Third-party tools are discussed in a more general way.

Using the Show Interfaces Command to Troubleshoot Serial Lines

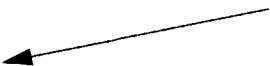
One of Cisco's most useful built-in diagnostic tools is the **show interfaces EXEC** command. The specific information displayed depends on the interface type being examined (serial, Ethernet, token ring, or FDDI) and the type of encapsulation to which the network is connected (such as X.25 or SMDS). This discussion focuses on information in the serial version of the display and on outlining the specific fields used to diagnose WAN/serial line connectivity problems.

Figure 7-1 illustrates the **show interfaces serial number** command output for a basic serial interface.

Note: Throughout this chapter, when the **show interfaces serial number** command is discussed, the required interface *number* specification is implied, although it is generally dropped unless specified in a command listing.

The interface depicted is not running packet-switched software. The fields presented in this display are detailed in your *Router Products Configuration and Reference* publication.

Interface status line



```
Serial 0 is up, line protocol is up
Hardware is MCI Serial
Internet address is 131.108.156.98, subnet mask is 255.255.255.240
MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
Encapsulation HDLC, loopback not set, keepalive set (10 sec)
Last input 0:00:00, output 0:00:00, output hang never
Last clearing of "show interface" counters never
Output queue 0/40, 0 drops; input queue 0/75, 0 drops
Five minute input rate 9000 bits/sec, 16 packets/sec
Five minute output rate 9000 bits/sec, 17 packets/sec
 50806 packets input, 5841604 bytes, 0 no buffer
  Received 42757 broadcasts, 0 runts, 21 giants
 146124 input errors, 87243 CRC, 58857 frame, 0 overrun, 0 ignored, 3 abort
 298821 packets output, 15669598 bytes, 0 underruns
   0 output errors, 0 collisions, 141 interface resets, 0 restarts
   7 carrier transitions
```

Figure 7-1 Display Output for HDLC Version of Show Interfaces

Information for analyzing the key diagnostic fields in this display is provided in the sections that follow. Several important diagnostic fields are presented in Figure 7-1 that help isolate WAN and serial line problems. These include:

- Line and protocol status
- Number of input errors
- Number of output drops
- Number of input drops
- Number of interface resets
- Number of carrier transitions
- WAN-specific diagnostic fields
- Traffic rate/line utilization
- Encapsulation type

Deciphering Serial Line Status Diagnostics

There are five possible “problem” states that can be identified in the interface status line (see Figure 7-1) of the **show interfaces serial** display:

- Serial (n) is down, line protocol is down
- Serial (n) is up, line protocol is down
- Serial (n) is up, line protocol is up (looped)
- Serial (n) is up, line protocol is down (disabled)
- Serial (n) is administratively down, line protocol is down

The causes and suggested actions associated with each of these conditions is summarized in Table 7-1.

Table 7-1 Show Interfaces Serial Status Line Problem States

Status Line Display Output (Symptom)	Possible Causes and Suggested Actions
Serial 0 is down, line protocol is down	<p>Possible Causes:</p> <ol style="list-style-type: none"> 1. Faulty cabling 2. Incorrect applique type 3. Router hardware failure <p>(This status indicates that the router is not detecting a carrier signal—DCD is not active).</p> <p>Suggested Actions:</p> <p><i>Step 1:</i> Inspect LED indicators for failure conditions (refer to your hardware installation and maintenance publication for LED descriptions).</p> <p><i>Step 2:</i> Insert breakout box; check all control leads</p> <p><i>Step 3:</i> Install correct applique if wrong</p> <p><i>Step 4:</i> Swap faulty part(s)</p>
Serial 0 is up, line protocol is down (DTE mode)	<p>Possible Causes:</p> <ol style="list-style-type: none"> 1. Router hardware failure 2. Remote router down or misconfigured 3. Failed local CSU/DSU 4. Failed remote CSU/DSU 5. Failure at switch (for packet-switched network only) <p>Suggested Actions:</p> <p><i>Step 1:</i> Put modem or CSU/DSU in local loopback mode and determine if line protocol comes up in show interfaces output.</p> <p><i>Step 2:</i> Enable debug serial-interface</p> <p><i>Step 3:</i> If the line protocol does not come up in local loopback mode (keepalive counter does not increment), router hardware problem is likely; swap router interface hardware.</p> <p><i>Step 4:</i> If the keepalive counter increments and line comes up, problem is <i>not</i> in local router. Troubleshoot serial line per loopback and ping tests described later in this chapter.</p>
Serial 0 is up, line protocol is down (DCE mode)	<p>Possible Causes:</p> <ol style="list-style-type: none"> 1. DTE device does not support (or is not set up for) SCTE mode 2. Router hardware failure 3. Failed remote CSU/DSU 4. No clockrate command <p>Suggested Actions:</p> <p><i>Step 1:</i> Set DTE device to SCTE mode if possible.</p> <p><i>Step 2:</i> Set DCE applique to SCT mode if DTE does not support SCTE.</p>

Status Line Display Output (Symptom)	Possible Causes and Suggested Actions
	<p><i>Step 3:</i> If protocol is still down, then there is a possible hardware failure or cabling problem. Insert a breakout box and observe leads.</p> <p><i>Step 4:</i> Replace faulty part(s) as necessary.</p>
Serial 0 is up, line protocol is looped	<p>Possible Causes:</p> <ol style="list-style-type: none"> 1. Loop exists in circuit 2. Routers on each end of the serial line came up simultaneously (system operates normally). This is normal behavior for systems running software prior to Software Release 8.3. <p>For routers running Software Release 8.3 and higher, the sequence number in the keepalive packet changes to a random number when a loop is initially detected. If the random number is returned over the link, a loop exists.</p> <p>Suggested Actions:</p> <p><i>Step 1:</i> Check router configuration using the write terminal command. Look for any instances of the loopback interface subcommand.</p> <p><i>Step 2:</i> If found, enter configuration mode and remove the loop using the no loopback interface subcommand for the appropriate interface.</p> <p><i>Step 3:</i> If not found, examine the DSU/CSUs to determine if they are configured in manual loopback mode. If they are, disable manual loopback. If not, contact the leased-line or other carrier service for line troubleshooting assistance.</p> <p><i>Step 4:</i> Reset one of the routers; inspect line status display; if protocol comes up, then Problem #2 is likely cause. No other action is needed.</p>
Serial 0 is up, line protocol is disabled	<p>Possible Causes:</p> <ol style="list-style-type: none"> 1. Hardware problem at the CSU/DSU 2. Bad router applique 3. Large number of input errors <p>Suggested Actions:</p> <p><i>Step 1:</i> Troubleshoot with serial analyzer and breakout box; look for hardware control leads toggling (CTS and DSR)</p> <p><i>Step 2:</i> Swap out bad hardware as required (DSU/CSU, switch, local or remote router).</p>
Serial 0 is administratively down, line protocol is down	<p>Possible Causes:</p> <ol style="list-style-type: none"> 1. Router configuration includes the shutdown interface subcommand for the interface. <p>Suggested Actions:</p> <p><i>Step 1:</i> Check router configuration for shutdown command.</p> <p><i>Step 2:</i> Remove command using the no shutdown interface subcommand.</p>

Basic Serial Diagnostic Fields

Five important error fields are provided in the **show interfaces serial** display output with respect to WAN and serial line troubleshooting:

- Number of input errors
- Number of output drops
- Number of input drops
- Number of interface resets
- Number of carrier transitions

Figure 7-2 illustrates the position of these information fields in the output of the **show interfaces serial** command.

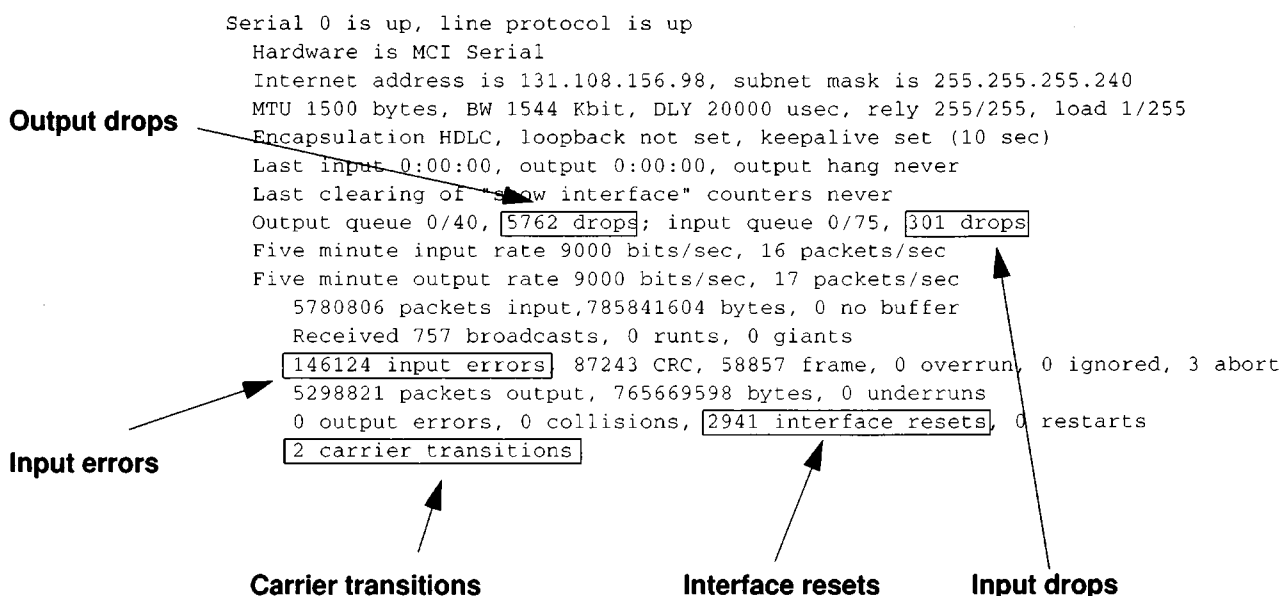


Figure 7-2 Show Interfaces Serial Diagnostic Field Locations

Although these fields are described in your *Router Products Configuration and Reference* publication, the following discussions provide some additional details and outline how you can use them to diagnose connectivity problems for WAN and serial line connections.

Evaluating Input Errors

When input errors appear in the **show interfaces serial** output, you must consider several possibilities in order to determine the source of those errors. Most likely problems are summarized in the list of possible causes that follows.

Note: Any input error value for CRC errors, framing errors, or aborts above one percent of the total interface traffic suggests some kind of hardware problem that should be isolated. Problem can be any element in the serial link, including a noisy line.

Symptom:

Increasing number of input errors in excess of one percent of total interface traffic.

Possible Causes:

1. Dirty serial line
2. Incorrect clocking configuration
3. Bad cable
4. Bad CSU/DSU
5. Bad MCI/SCI card
6. Bad applique
7. Data converter being used

Note: Cisco strongly recommends against using data converters when interconnecting to a WAN or serial network with a router.

Suggested Actions:

Step 1: Use a serial analyzer to isolate the source of the errors. If errors are detected by the serial analyzer, then some device external to the router is the source (bad hardware or clock mismatch).

Step 2: Use the loopback and ping tests described later in this chapter to isolate specific problem source.

Table 7-2 details the meaning of CRC errors, framing errors, and aborts. These fields appear in the display shown in Figure 7-2.

Table 7-2 Meaning of Key Input Errors for Serial Line Troubleshooting

Input Error Type (Field Name)	Possible Causes and Suggested Actions
CRC Errors (CRC)	<p>Meaning: CRC calculation does not pass.</p> <p>Possible Causes:</p> <ol style="list-style-type: none">1. Dirty serial line, noise.2. Clocking skews; serial cable is too long (greater than 50 feet, or 25 feet for T1 link); cable from CSU/DSU to router is not shielded; SCTE/ Terminal timing not enabled on DSU; CSU line clock incorrectly configured.

Input Error Type (Field Name)	Possible Causes and Suggested Actions
Framing Errors (frame)	<p data-bbox="561 241 808 269">Suggested Actions:</p> <p data-bbox="561 275 1411 340">Step 1: Ensure that line is clean enough for transmission requirements; shield cable if needed.</p> <p data-bbox="561 357 1411 422">Step 2: Ensure that all devices are properly configured to use common line clock.</p> <hr/> <p data-bbox="561 461 683 489">Meaning: Detected packet does not end on an eight-bit byte boundary</p> <p data-bbox="561 545 773 573">Possible Causes:</p> <ol data-bbox="561 579 1422 890" style="list-style-type: none"> 1. Dirty serial line, noise 2. Clocking jitter; improperly designed cable; serial cable is too long (greater than 50 feet, or 25 feet for T1 link); cable from CSU/DSU to router is not shielded 3. Clocking skews; serial cable is too long (greater than 50 feet, or 25 feet for T1 link); cable from CSU/DSU to router is not shielded; SCTE/ Terminal timing not enabled on DSU; CSU line clock incorrectly configured; one of the clocks is configured for local clocking. <p data-bbox="561 907 808 935">Suggested Actions:</p> <p data-bbox="561 942 1411 1006">Step 1: Ensure that line is clean enough for transmission requirements; shield cable if needed.</p> <p data-bbox="561 1024 1411 1088">Step 2: Ensure that all devices are properly configured to use common line clock</p> <hr/>
Aborted Transmission (abort)	<p data-bbox="561 1134 683 1162">Meaning: Illegal sequence of one bits</p> <p data-bbox="561 1218 773 1246">Possible Causes:</p> <ol data-bbox="561 1252 1422 1403" style="list-style-type: none"> 1. Packet terminated in middle of transmission; typical cause is an interface reset 2. Hardware problem—bad circuit, bad CSU/DSU, bad sending interface on remote router <p data-bbox="561 1420 808 1448">Suggested Actions:</p> <p data-bbox="561 1455 1162 1483">Step 1: Check hardware at both ends of the link.</p> <p data-bbox="561 1500 1101 1528">Step 2: Swap faulty equipment as necessary.</p> <hr/>

Evaluating Output Drops

Output drops appear in the **show interfaces serial** output when no buffers are available and the system is attempting to hand off a packet to a transmit buffer. The output drops count is illustrated in Figure 7-2.

Symptom:

Increasing output drops

Possible Cause:

Overused serial interface

Suggested Actions:

Step 1: Turn off fast switching on affected interfaces.

Step 2: Change output hold queue size.

Step 3: Implement priority queuing.

Note: Output drops are acceptable under certain conditions. For instance, if a link is known to be overused (with no opportunity or way to remedy the situation), it is often considered preferable to drop packets than to hold them. This is true for protocols (such as TCP/IP and Novell IPX) that support flow control and that can retransmit data. However, some protocols, such as DECnet, are sensitive to dropped packets and have poor accommodations for retransmission.

Evaluating Input Drops

Input drops appear in the **show interfaces serial** command when too many packets from that interface are still being processed in the system. The input drops count is illustrated in Figure 7-2.

Symptom:

Increasing number of input drops

Possible Cause:

Overutilized interface

Note: Input drop problems are typically seen when traffic is being routed between faster interfaces (such as Ethernet, FDDI, and Token Ring) and serial interfaces. When traffic is light, there is no problem. As traffic rates increase, backups start occurring. Routers by design drop packets during these congested periods.

Suggested Actions:

Increase input hold queue size.

Evaluating Interface Resets

Interface resets appear in the **show interfaces serial** command when they result from missed keepalives. The interface resets count is illustrated in Figure 7-2.

Symptom:

Increasing interface resets

Possible Causes:

1. Bad line causing carrier (DCD) transitions.
2. Lack of buffer availability (prior to Software Release 8.3) and congestion on link (typically associated with output drops).
3. Possible hardware problem at CSU/DSU or switch.

Suggested Actions:

When looking at interface resets, you must examine other fields as well to determine possible causes. Assuming an increase in interface resets is being recorded, examine the following fields:

- Step 1:** Check the carrier transitions field in the **show interfaces** display. If carrier transitions are high while interface resets are being registered, the problem is likely to be a bad link or bad CSU/DSU. Contact your leased line/carrier service and swap faulty equipment as necessary.
- Step 2:** Examine the input errors field in the **show interfaces** display. If input errors are high while interface resets are increasing, the problem also is probably a bad link or DSU/CSU. Contact your leased line/carrier service and swap faulty equipment as necessary.
- Step 3:** Examine the output drops field in the **show interfaces** display. If your system is running a version of software prior to Software Release 8.3, and a high number of output drops are registered along with interface resets, upgrade to Software Release 8.3 or higher.

Evaluating Carrier Transitions

Carrier transitions appear in the **show interfaces serial** command whenever there is an interruption in the carrier signal; for example, when there is an interface reset at the remote end of a link. The carrier transition count is illustrated in Figure 7-2.

Symptom:

Increasing carrier transitions

Possible Causes:

1. Line interruptions due to external source (examples: lighting strikes somewhere along the network; physical separation of cabling; Red or Yellow T1 alarms)
2. Faulty switch, DSU, or router hardware

Suggested Actions:

- Step 1:** Check hardware at both ends of the link (attach breakout box or serial analyzer and test to determine source of problems).
- Step 2:** If analyzer or breakout box are unable to identify any external problems, check router hardware.
- Step 3:** Swap faulty equipment as necessary.

WAN-Specific Diagnostic Fields

In addition to the basic diagnostic fields provided in all **show interfaces serial** displays, additional information is provided with different WAN encapsulations (such as X.25, SMDS, or Frame Relay). These displays are discussed in detail in your *Router Products Configuration and Reference* publication.

The discussion that follows is provided to illustrate the kind of supplemental information available through the **show interfaces serial** command for troubleshooting WAN connections. It focuses on displays provided with the X.25-specific **show interfaces serial** display.

Example: X.25 Show Interfaces Diagnostic Fields

The general serial diagnostics fields discussed in the preceding section apply to X.25 and other WAN connections as well. In addition, five additional error fields are provided in the **show interfaces serial** display output for X.25 internetworks. These fields provide the following information:

- Number of rejects (REJs)
- Number of SABMs
- Number of Receiver Not Ready (RNR) flags
- Number of framing errors (FRMRs)
- Number of restarts (RESTARTS)
- Number of disconnects (DISCs)

Figure 7-3 illustrates the position of each of these and other useful information fields in the X.25 version of the display output for the **show interfaces serial** command.

```

Serial 1 is up, line protocol is up
Hardware is MCI Serial
Internet address is 192.126.139.1, subnet mask is 255.255.255.0
MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 254/255, load 2/255
Encapsulation X25, loopback not set
LAPB state is CONNECT, TI 500, N1 12040, N2 20, K 7
VS 5, VR 0, RCNT 0, Remote VR 3, Retransmissions 0
IFRAMES 2483134/2293869 RNRs 0/17 REJs 209/3 SABMs 2/0 FRMRs 0/10 DISCs 0/0
x25 address 39802154724800, state R1, modulo 8, file 0, timer 0, nvc 2
Window size: input 7, output 7, Packet size: input 1024, output 1024
Timers: T20 180 T21 200 T22 180 T23 180 TH 0
Channels: Incoming 1-127 Two-way 1-127 Outgoing 1-127
RESTARTs 31/14 CALLs 14+2/0+0/0+0 DIAGs 0/0
Last input 0:00:00, output 0:00:00, output hang never
Last clearing of "show interface" counters never
Output queue 0/40, 0 drops; input queue 0/75, 0 drops
Five minute input rate 25000 bits/sec, 18 packets/sec
Five minute output rate 16000 bits/sec, 19 packets/sec
4856022 packets input, 9451019395 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants
7 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 7 abort
4236887 packets output, 9121574650 bytes, 0 underruns
0 output errors, 0 collisions, 0 interface resets, 0 restarts

```

Number of rejects points to the line: `RESTARTs 31/14`

RNR count points to the line: `RNRs 0/17`

Number of restarts points to the line: `RESTARTs 31/14`

SABM count points to the line: `SABMs 2/0`

Number of framing errors points to the line: `FRMRs 0/10`

Number of disconnects points to the line: `DISCs 0/0`

Figure 7-3 Display Output for X.25 Version of Show Interfaces

Note: If any of these errors are increasing and represent more than 0.5 percent of the IFRAMES measured, there is probably a problem somewhere in the X.25 network. There should always be at least one SABM; however, if there are more than 10, the packet switch probably is not responding.

Symptom:

Recorded REJs, RNRs, FRMRs, RESTARTS, or DISCs in excess of 0.5 percent of IFRAMES

Possible Causes:

1. Faulty switch
2. Bad cabling
3. Bad CSU/DSU
4. Failed router hardware

Suggested Actions:

- Step 1:** Enable **debug x25-events** diagnostic command; evaluate "Cause and Diagnostic" information provided. Refer to Appendix A, "X.25 Cause and Diagnostics" for details concerning the meaning of the codes generated.
- Step 2:** Check hardware at both ends of the link using a serial analyzer. Use the analyzer to determine whether the SABMs are being responded to with UAs.

Step 3: If analyzer cannot identify any external problems, check router hardware.

Step 4: Swap faulty equipment as necessary.

Using the Show Controllers Command to Troubleshoot Serial Lines

Another important diagnostic tool is the **show controllers** command. For serial interfaces, use the **show controllers mci** command. If the electrical interface is displayed as “UNKNOWN” (instead of V.35, RS-449, or some other electrical interface type), then a likely problem is either a bad applique or a problem with the card’s internal wiring. In addition, the corresponding display for the **show interfaces serial** command will show the interface and line protocol as down in the interface status line. Figure 7-4 illustrates this kind of display output for the **show controllers mci** command.

**Electrical interface identified as type
UNKNOWN, suggesting a hardware failure.**

```
MCI 1, controller type 1.1, microcode version 1.8
 128 Kbytes of main memory, 4 Kbytes cache memory
16 system TX buffers, largest buffer size 1520
Restarts: 0 line down, 0 hung output, 0 controller error
Interface 0 is Ethernet1, station address 0000.0c00.3b09
 22 total RX buffers, 9 buffer TX queue limit, buffer size 1520
Transmitter delay is 0 microseconds
Interface 1 is Serial2, electrical interface is UNKNOWN
 22 total RX buffers, 9 buffer TX queue limit, buffer size 1520
Transmitter delay is 0 microseconds
High speed synchronous serial interface
Interface 3 is Serial3, electrical interface is V.35 DTE
 22 total RX buffers, 9 buffer TX queue limit, buffer size 1520
Transmitter delay is 0 microseconds
High speed synchronous serial interface
```

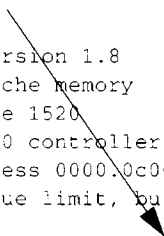


Figure 7-4 Example Display Output of Show Controllers MCI Command

Using Debug Commands to Troubleshoot Serial Lines

The output from **debug** commands provides diagnostic information concerning a variety of internetworking events relating to protocol status and network activity in general. The following are some **debug** commands that are useful when troubleshooting serial and WAN problems.



Caution: Throughout this and other chapters, the use of **debug** commands is suggested for obtaining information about network traffic and router status. Use these commands with great care. In general, it is recommended that these commands only be used under the direction of your router technical support representative when troubleshooting specific problems. Enabling debugging can disrupt operation of the router when internets are experiencing high load conditions. When you finish using a **debug** command, remember to disable it with its specific **undebug** command or with the **undebug all** command.

More information is provided in Chapter 10, “Debug Command Reference.” In addition, when a specific **debug** command is recommended for a particular situation, appropriate reminders are included in the various scenarios and symptom modules.

- **debug serial-interface**—Verifies whether HDLC keepalives are incrementing; if not, a possible timing problem exists on the interface card or in the network.
- **debug x25-events**—Detects X.25 events, such as the opening and closing of switched virtual circuits (SVCs). The resulting “Cause and Diagnostic” information will be included with the event report. Refer to Appendix A, “X.25 Cause and Diagnostics,” for more information concerning this output.
- **debug lapb**—Obtains LAPB or Level 2 X.25 information.
- **debug arp**—Indicates whether the router is sending information about or learning about routers on the other side of the WAN cloud. Use this command when some nodes on a TCP/IP network are responding, but others are not. It shows whether or not the router is sending or receiving ARPs.
- **debug frame-relay-lmi**—Obtains LMI information for determining whether a Frame Relay switch and router are sending/receiving LMI packets.
- **debug frame-relay-events**—Determines whether exchanges are occurring between a router and a Frame Relay switch.
- **debug serial-packet**—Shows SMDS packets being sent and received. This display also prints out necessary error messages to indicate why a packet was not sent or was received erroneously. For SMDS, dumps the entire SMDS header and some payload data when an SMDS packet is transmitted or received.

Special Serial Line Tests

In addition to the basic diagnostic capabilities provided with Cisco internetworking systems, there are a variety of supplemental tools and techniques that can be used to determine the conditions of cables, switching gear, modems, hosts, and remote internetworking hardware. Although complete discussions of these tools are beyond the scope of this publication, some hints about using these alternative tools are provided here. Consult your manufacturer’s manuals (CSU/DSUs, serial analyzers, etc.) for more information and specific instructions.

CSU/DSU Local and Remote Loopback Tests

If the output of the **show interfaces serial** command indicates that the serial line is up but the line protocol is down, use the DSU loopback tests to determine the source of the problem. Perform the local loop test first.

Note: These tests are generic in nature and assume attachment of the internetworking system to a CSU/DSU. However, the test is essentially the same for attachment to a multiplexer with built-in CSU/DSU functionality. Since there is no concept of a “loopback” in an X.25 PSN environment, loopback tests do not apply to X.25 networks.

CSU/DSU Local Loopback Tests for HDLC Links

The following discussion outlines a general procedure for performing loopback tests in conjunction with built-in Cisco system diagnostic capabilities.

- Step 1:** Place the DSU in local loop mode. In local loop mode, the use of the line clock (from the T1 service) is terminated and the DSU is forced to use the local clock.
- Step 2:** Use the **show interface serial EXEC** command to determine if the line status changes state (from “line protocol is down” to “line protocol is up (looped)”) or if it remains down.

If the **show interfaces serial** status line changes state when the CSU/DSU is in local loopback mode, it suggests that the problem is external to the router. If the status line does not change state, there is a possible problem in the router, connecting cable, or DSU.

- Step 3:** Use the **debug serial-interface EXEC** command to further evaluate the condition of the serial line.

Before enabling the CSU/DSU’s local loop test, when the line protocol is *down*, the **debug serial-interface** output will indicate that keepalives are not incrementing.

Placing the CSU/DSU into local loop mode should cause the keepalives to begin incrementing. Specifically, the values for *mineseen* and *yourseen* keepalives will increment every 10 seconds. This information will appear in the **debug serial-interface** output. If they do not increment, there may be a timing problem on the interface card or on the network (refer to the subsequent section, “Troubleshooting Clocking Problems,” for additional information).

Note: If the **show interfaces serial** output indicates “Serial (n) is up, line protocol is down” while in this mode, the problem is likely to be a bad cable, bad CSU/DSU, or a clocking problem.

CSU/DSU Remote Loopback Tests for HDLC Links

If you are able to determine that the local hardware is functioning properly but cannot establish connections over the serial link, try using the remote loopback test that follows to isolate the problem cause.

Note: Assuming HDLC encapsulation, this remote loopback test assumes that the preceding local loop test was performed immediately before this test.

Step 1: Put the CSU/DSU into remote loopback.

Step 2: Using the **show interfaces serial EXEC** command, determine whether the line protocol *remains* up, with the status line indicating “Serial *n* is up, line protocol is up (looped).”

If the line protocol remains up in remote loopback mode, the problem is probably at the far end of the serial connection. Perform both tests (local and remote) at the far end to determine the problem source.

If the line status displayed via **show interfaces serial** toggles to “Line protocol is down” when remote loopback mode is activated, the problem is in the local CSU/DSU or somewhere in the line to the CSU/DSU. Try replacing the local CSU/DSU. If this does not resolve the problem, contact your WAN network manager or the WAN service organization.

Extended Ping Tests

One of the most useful tests available on Cisco internetworking systems (as well as on many host systems) is the *ping* function. In the TCP/IP world, this diagnostic tool also is known as the ICMP Echo Request.

Note: This test is particularly useful when high levels of input errors are being registered in the **show interface serial** display (refer to Figure 7-2).

Cisco internetworking systems provide a mechanism to automate the sending of many ping packets in sequence. Figure 7-5 illustrates the menu used to specify extended ping options. This example specifies 20 successive pings; however, when testing the components on your serial line, you should specify a much larger number, such as 1000 pings.

```

Betelgeuse# ping
Protocol [ip]:
Target IP address: 129.44.12.7
Repeat count [5]: 20
Datagram size [100]: 64
Timeout in seconds [2]:
Extended commands [n]: yes
Source address:
Type of service [0]:
Set DF bit in IP header? [no]:
Data pattern [0xABCD]: ffff
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 20, 64-byte ICMP Echos to 129.44.12.7, timeout is 2 seconds:
Packet has data pattern 0xFFFF
!!!!!!!!!!!!!!!!!!!!
Success rate is 100 percent, round-trip min/avg/max = 1/3/4 ms

```

Ping count specification

Extended command selection option

Data pattern specification

Figure 7-5 Extended Ping Specification Menu

In general, perform serial line **ping** tests as follows:

- Step 1:** Put CSU/DSU into local loopback mode.
- Step 2:** Configure the extended **ping** to send all zeros or all ones (Figure 7-5 illustrates specification of all ones in the “Data pattern [0xABCD]:” option field).
- Step 3:** Examine the serial interface statistics and determine whether input errors have increased. If input errors have not increased, the local hardware (DSU, cable, router interface card, and applique) are likely to be good.

Assuming this test sequence was prompted by the appearance of a large number of CRC and framing errors, a likely suspect is a clocking problem. Check the CSU/DSU for a timing problem. Refer to the discussion “Troubleshooting Clocking Problems” that follows immediately after this section.
- Step 4:** If you determine that the clocking configuration is correct and operating properly, put the CSU/DSU into remote loopback mode.
- Step 5:** Repeat the **ping** test and look for changes in the input error statistics.
- Step 6:** If input errors increase, there is either a problem in the WAN/serial line or on the CSU side of the CSU/DSU. Contact the WAN service provider and swap the CSU/DSU. If problems persist, consult your router technical support representative.

Troubleshooting Clocking Problems

Clocking conflicts in serial connections can lead to either chronic loss of connection service or generally degraded performance. The following discussion addresses four issues regarding clocking problems:

- Clocking problem causes
- Methods for detecting clocking problems
- General steps to isolate the source of clocking conflicts
- Possible remedies

Clocking Problem Causes

In general, clocking glitches in serial WAN interconnections can be attributed to one of the four basic causes:

- Incorrect DSU configuration
- Incorrect CSU configuration
- Cables out of specification (longer than 50 feet or unshielded)
- Noisy or poor patch panel connections

Detecting Clocking Problems

To detect clocking conflicts on your serial interface, look for input errors as follows:

- Step 1:** Use the **show interfaces serial** command on the routers at both ends of the link.
- Step 2:** Examine the display output for CRC and framing errors.
- Step 3:** If either of these steps indicate errors exceeding an approximate range of 0.5 to 1.0 percent of traffic on the interface, clocking problems are likely to exist somewhere in the WAN.
- Step 4:** Isolate the source of the clocking conflicts as outlined in the next procedure.

Isolating Clocking Problems

Once you have determined that clocking conflicts are the most likely cause of input errors, the following general steps will help you isolate the source of those errors:

- Step 1:** Perform a series of loopback and ping tests, both local and remote, as described in procedures that precede this section (“CSU/DSU Local and Remote Loopback Tests” and “Extended Ping Tests”).
- Step 2:** Determine which end of the connection is the source of the problem, or if the problem is in the line. In “local” loopback mode, run different patterns and sizes in the ping tests (for instance 1500 bytes). Using a single pattern and packet size may not force errors to materialize, particularly when a serial cable to the router or CSU/DSU is the problem.
- Step 3:** Perform the **show interfaces serial EXEC** command and determine whether input errors counts are increasing and where they are accumulating.
- If input errors are accumulating on both ends of the connection, clocking of the CSU is the likely problem.
- If only one end is experiencing input errors, there is likely to be a DSU clocking or cabling problem.
- If you see aborts on one end, it suggests that the *other* end is sending bad information or that there is a line problem.

Note: Always refer back to the **show interfaces serial** display output and log any changes in error counts or note if error count does not change.

Suggested Clocking Problem Remedies

Table 7-3 outlines suggested remedies for clocking problems, based on the source of the problem.

Table 7-3 Sources of and Suggested Remedies for Clocking Problems

Clocking Problem	Suggested Action
Incorrect CSU configuration	Step 1: Determine whether the CSUs at both ends are in agreement regarding the clock source (local or line). Step 2: Configure both to agree if not already correctly configured (usually the line is the source).
Incorrect DSU configuration	Step 1: Determine whether the DSUs at both ends have SCTE (terminal timing) enabled. Step 2: Enable SCTE if not already correctly configured. (For any interface that is connected to a line of 128 Kbps or faster, SCTE must be enabled).
Cable to router out of specification	Step 1: Use shorter cable if longer than 50 feet. Step 2: Replace with shielded cable.

Adjusting Buffers to Ease Overutilized Serial Links

Excessively high bandwidth utilization results in reduced overall performance and possibly intermittent failures. For example, DECnet file transmissions may be failing due to packets being dropped somewhere in the network. If the situation is bad enough, you *must* add bandwidth. However, adding bandwidth may not be necessary or immediately practical. One way to resolve marginal serial line overutilization problems is to control how the router uses data buffers.

You have three options to control how buffers are used:

- Adjust parameters associated with system buffers
- Specify the number of packets held in input or output queues (called *hold queues*)
- Prioritize how traffic is queued for transmission (also called *priority output queuing*)

The configuration commands associated with these options are fully described in the *Router Products Configuration and Reference* publication.

The following discussion focuses on identifying situations in which these options are likely to apply and defining how you can use these options to help resolve connectivity and performance problems in serial/WAN interconnections. Commands are discussed as appropriate.

Managing System Buffers

There are two general buffer types on Cisco routers. These are referred to as *hardware* buffers and *system* buffers. Only the system buffers are directly configurable by system administrators.

The hardware buffers are specifically used as the receive and transmit buffers associated with each interface and (in the absence of any special configuration) are dynamically managed by the system software itself.

Note: When you use the priority queuing mechanism to control traffic flow, the number of transmit buffers for the associated interface is implicitly set to one. This is done because the use of this feature forces the use of the system buffers instead of hardware buffers.

The system buffers are associated with the main system memory and are allocated to different size memory blocks. As of software release 8.3, these buffers can be configured using the **buffers** global configuration command.

The most useful parameters to set with the **buffers** command are **min-free**, **initial**, and **max-free**. Refer to your *Router Products Configuration and Reference* publication for more information concerning this command and its options.

Note: The **buffer** command's **max-free** option is especially important when also using the **hold-queue** interface subcommand. The **max-free** number must be set large enough to ensure that enough buffer space is available to be supplied on demand.

The following procedure outlines a process for implementing system buffer changes:

Step 1: Determine whether adjusting system buffers may be useful. Use the **show interfaces** command and look for a high number of output drops on serial interfaces.

Step 2: Prepare the router for switching to using system buffers. Do this by disabling fast switching.

Use the **write terminal** command to determine what protocols are being run on the router and whether there are explicit instances of the various **route-cache** interface subcommands (such as **ip route-cache**). Disable these with the **no** versions of the command.

Since fast switching is enabled by default, you also must explicitly disable fast switching with these commands (such as **no ip route-cache**) for all protocols on affected serial interfaces.

Step 3: Identify which buffers are likely to be experiencing problems. Use the **show buffers** command and look for misses in the system buffers listed. Table 7-4 lists the relative size of each system buffer type.

Table 7-4 Range of Packet Sizes Held in System Buffers

Buffer Size Indicated	Packet Size Range Buffered	Typical Traffic Related to Drops
Small	64 to 104 bytes	Virtual Terminal (DEC LAT, Telnet)
Middle	105 to 600 bytes	Novell IPX
Big	601 to 1524 bytes	File Transfer (FTP)
Large	1525 to 5024 bytes	File Transfer (Token Ring)
Huge	5025 to 12024 bytes	Fragmented packets destined for a router (routing updates associated with large internets)

Step 4: Add a **buffers** global configuration command as appropriate, tailored to allow enough buffers to prevent misses for protocols where misses are of particular concern.

Note: When misses are indicated in the **show buffers** command display, input or output drops are likely. Two environments that are particularly sensitive to output drops are DECnet and large Novell IPX networks. In DECnet, when one packet is dropped, the entire transmission must be resent. This can be very inefficient and can kill a network if large file transfers must be repeatedly resent. In large IPX networks, excessive, simultaneous SAP updates can take up all buffers and slow normal serial traffic.

The following are guidelines for specifying the number of buffers when you see misses (output drops) occurring:

- As a starting point, set the minimum number of free buffers to 20 small buffers, 10 medium buffers, and 10 big buffers. Vary these values depending on the kind of misses occurring.
- For Novell, settings for the minimum and maximum number of buffers depends on the number of servers that the router can access. Determine how many packets are likely to be generated via simultaneous SAP updates. Set the minimum number of buffers to 5 more than the anticipated number of packets generated, and set the maximum to at least 150 more than the minimum free buffer setting.

Note: In addition to buffer management, Cisco recommends using *SAP filtering* across serial interconnections for large Novell IPX networks. In general, large, unfiltered networks tend to congest serial traffic. For example, a Novell network of 300 servers can generate enough traffic to continuously fill a 56-Kbps line for more than one second. SAP filtering and increasing the SAP update times can relieve some congestion. SAP filters should be applied *before* modifying the number of system buffers. If misses resulting from congestion persist, calculate the number of buffers required to eliminate misses and apply changes with the **buffers** global configuration command.

- When you set the minimum number of free buffers, the effect is to set an internal flag when the system experiences demand for the specified buffer type in excess of the indicated minimum. When the CPU is able to attend this buffer management, more buffers are built to maintain the specified minimum free buffer level.

Implementing Hold Queue Limits

Hold queues are buffers used by the router for each interface to store outgoing or incoming packets. Use the **hold-queue** interface subcommand to increase the number of data packets queued before the router will drop packets.

Note: The **hold-queue** command is useful only in process-switched mode. Fast switching must be disabled on the interface to use this function.

Use this command to prevent packets from being dropped and to improve serial link performance under the following conditions:

1. You have an application that cannot tolerate drops and the protocol is able to stand longer delays. DECnet is an example of a protocol that meets both criteria. LAT does not since it does not tolerate delays.
2. The interface is very slow (low bandwidth and/or anticipated utilization is likely to sporadically exceed available bandwidth).

Note: When you increase the number specified for an output hold queue, you must increase the number of system buffers. The value used depends on the size of the packets associated with the traffic anticipated for the network.

Using Priority Queuing to Reduce Bottlenecks

Priority queuing is a list-based control mechanism that allows network administrators to prioritize traffic transmitted into networks on an interface-by-interface basis. In a manner that is analogous to Cisco's access list traffic control mechanisms, priority queuing involves two steps:

Step 1: Create a priority list, by protocol type and level of priority.

Step 2: Assign the priority list to a specific interface.

Both these steps use versions of the **priority-list** global configuration command (with the keywords **protocol** and **interface**, as appropriate). In addition, further traffic control can be applied by referencing **access-list** global configuration commands from **priority-list** specifications. Refer to your *Router Products Configuration and Reference* publication for examples of defining priority lists and details about command syntax associated with priority queuing.

Note: Do not use priority queuing and hold queues at the same time. Priority queuing automatically creates four hold queues of varying size. This overrides any hold queue specification included in your configuration.

Use priority queuing to prevent packets from being dropped and to improve serial link performance under the following conditions:

1. When the interface is slow, there are a variety of traffic types being transmitted, and you want to improve terminal traffic performance.
2. If you have a serial link that is intermittently experiencing very heavy loads (such as file transfers occurring at specific times), you can use priority lists to select which types of traffic should be discarded at high traffic periods.

In general, start with the default number of queues (altered with the **queue-limit** keyword option of the **priority-lists** global configuration command) when implementing priority queues. After enabling priority queuing, monitor output drops with the diagnostic command **show interfaces serial**. If you notice that output drops are occurring in the traffic queue you have specified to be high priority, increase the number of packets that can be queued.

Note: When bridging DEC LAT traffic, your router cannot drop any packets, or LAT will not function (that is, sessions will crash). Experience suggests that a high priority queue depth of about 100 (specified with the **queue-limit** keyword) is a typical working value when your router is dropping output packets and the serial lines are subjected to about 50 percent bandwidth utilization. If the router is dropping packets and is at 100 percent utilization, you need another line. Another tool to relieve congestion when bridging DEC LAT is *LAT compression*. You can implement LAT compression with the interface subcommand **bridge-group group lat-compression**.

WAN and Serial Line Connectivity Symptoms

The symptom modules that follow pertain to serial and WAN problems. Unless otherwise indicated, each module is presented as a set of general problems applying to all WAN types (such as X.25, point-to-point serial, SMDS, or Frame Relay). Where there are special considerations associated with a specific network type, notes are included.

WAN connectivity symptoms discussed in this section include the following:

- Intermittent connectivity
- Connections die as load increases
- Connections die at a particular time of day
- Connections die after a certain amount of normal operation
- Users cannot connect to resources over a new HDLC link
- Users cannot connect to resources over a new X.25 link
- Users cannot connect to resources over a new Frame Relay link
- Users cannot connect to resources over a new SMDS link
- Specific users cannot connect to resources over the WAN

Intermittent Connectivity

Symptom: Intermittent connectivity implies that connections randomly come and go. Connections can be made between some nodes, while others nodes cannot connect—with no apparent reason.

Possible Causes and Suggested Actions

Table 7-5 outlines possible causes of intermittent connectivity in serial and WAN interconnections.

Table 7-5 Causes and Actions for WAN Intermittent Connectivity

Possible Cause	Suggested Action
Faulty interface card or cable	<p>Step 1: Check status of interface with show interface serial and show controller commands.</p> <p>Step 2: Look for line down condition and version level.</p> <p>Step 3: Upgrade microcode (firmware) if current code is older than 1.7.</p> <p>Step 4: Swap card or cable if nonoperational.</p>
Bad CSU/DSU	<p>Step 1: Check for input errors in show interface serial output.</p> <p>Step 2: Replace modem.</p> <p>Step 3: Observe behavior after modem change.</p>
Timing problem	<p>Step 1: Check CSU/DSU configuration for SCTE/Terminal timing enabled; enable if not already enabled.</p> <p>Step 2: If the CSU/DSU is properly configured, or if intermittent connectivity persists after enabling SCTE/Terminal timing on the CSU/DSU, verify that the correct network entity is generating the system clock; reconfigure nodes and modems if clocking is not properly configured.</p> <p>Step 3: If intermittent problems still persist, check cable length; if it is longer than 25 feet, you may need to invert clock on the MCI/SCI.</p>
Network generating invalid PRs (X.25)	<p>Step 1: Run diagnostics at the switch.</p> <p>Step 2: Swap switch hardware if necessary.</p>
Router generating invalid PRs (X.25)	<p>Step 1: Enable debug x25-events; examine cause and diagnostics output; consult Chapter 10 for more details.</p> <p>Step 2: Upgrade router software to 8.3(5) or higher revision. Software release 9.0(3) is recommended.</p>
Serial line congestion	<p>Step 1: Turn off fast switching.</p> <p>Step 2: Manage buffers (must have Software 8.3 or greater).</p> <p>Step 3: Apply priority list.</p>

Connections Die as Load Increases

Symptom: The typical symptom is that users complain incessantly about lost connections at peak periods. One example of this problem is in an environment featuring bridged DEC Local Area Transport (LAT) traffic and multiple routed protocols. Data entry input from users (or other application requests) may be getting buffered at the end of an already long input queue; eventually one end of the connection will time out.

Possible Causes and Suggested Actions

Table 7-6 outlines possible causes of load-related failures in serial and WAN interconnections.

Table 7-6 Causes and Actions for Load-Related WAN Problems

Possible Cause	Suggested Actions
Dirty serial line	<i>Step 1:</i> Determine if input errors are increasing. <i>Step 2:</i> If input errors appear, diagnose serial line per discussion earlier in this chapter.
Overutilized serial line	<i>Step 1:</i> If input errors do not appear, the problem is related to congestion. <i>Step 2:</i> Turn off fast switching. <i>Step 3:</i> Include an appropriate priority queuing configuration statement. <i>Step 4:</i> Adjust buffer size (8.3 or more recent software needed).

Connections Die at a Particular Time of Day

Symptom: This symptom is generally an example of connections dying under load. In this case, traffic on a serial link approaches saturation at specific times during the day, for instance, around 8:30 a.m., noon, and 5:30 p.m. The result is a loss of connections or the ability to make connections.

Possible Causes and Suggested Actions

Table 7-7 outlines possible causes of load-related failures associated with time-of-day problems in serial and WAN interconnections.

Table 7-7 Causes and Actions for Time-of-Day WAN Problems

Possible Cause	Suggested Action
Overutilized bandwidth	Step 1: Check applications being run, especially for very large file transfers scheduled at particular times of day.
	Step 2: If this is the case, set up a priority queue based on packet size to allow higher small-packet traffic (requires that the protocol allows flow control).
	Step 3: Rearrange file transfer timing by applications so that links are not overused during normal business hours.
	Step 4: Add bandwidth and consider dial backup over the new link for applications that are taking excessive bandwidth on existing links.
Unshielded cable runs are too close to EMI sources	Step 1: Check show interfaces serial display for input errors.
	Step 2: If loading is not the problem, and input errors are being registered, inspect cable runs for proximity to interference sources.
	Step 3: Relocate or shield cables if found to be near E-M source.

Connections Die After Some Period of Normal Operation

Symptom: When connections suddenly die after a period of relatively normal, error-free operation (and cannot be brought back), a hardware-related problem somewhere along the serial line is the likely culprit. However, there are other possible causes (discussed in the list that follows).

Possible Causes and Suggested Actions

Table 7-8 outlines possible causes of sudden failures that occur following essentially error-free operations over serial and WAN interconnections.

Table 7-8 Causes and Actions for “Sudden Death” WAN Problems

Possible Cause	Suggested Action
Hardware in the serial link died	<i>Step 1:</i> Use show interfaces serial to determine whether link is down. <i>Step 2:</i> If link is down, troubleshoot serial line per CSU/DSU loopback tests and ping tests described earlier or with serial analyzer.
Routing tables are incorrect	<i>Step 1:</i> If the link is up, use the appropriate show protocol route command (example: show ip route or show apple route). <i>Step 2:</i> Determine whether routes are correct; if not, look for source of bad routes (flapping link, backdoor bridge between the routed segments, or incorrect configuration of route redistribution between routing protocols).
Buffer misses or other software problem	<i>Step 1:</i> If the routing table is correct and the link is up, use the show buffers command to evaluate buffer status. <i>Step 2:</i> Refer to the discussion concerning buffer management earlier in this chapter for more information. <i>Step 3:</i> Modify buffers as necessary to prevent dropped connections. <i>Step 4:</i> Contact your technical support representative if all actions fail to resolve problem.

Users Cannot Connect to Resources over New HDLC Link

Symptom: When no traffic of any kind is passing through a newly installed router interconnecting broadcast networks via an HDLC point-to-point link, look for problems associated with the new installation.

Possible Causes and Suggested Actions

Table 7-9 outlines possible causes of connectivity problems that occur following new HDLC serial router installations.

Table 7-9 Causes and Actions for New Router Problems (Serial HDLC)

Possible Cause	Suggested Action
Link is dead	<i>Step 1:</i> Use show interfaces serial to determine whether link is down. <i>Step 2:</i> If link is down, troubleshoot serial line per CSU/DSU loopback tests and ping tests described earlier or with serial analyzer.
Keepalives not being received	<i>Step 1:</i> Use debug serial-inteface to determine status of keepalives. <i>Step 2:</i> If keepalives are not incrementing (probably will not be, given this line status), troubleshoot serial line per the loopback and ping tests discussed earlier in this chapter.

Users Cannot Connect to Resources over New X.25 WAN Link

Symptom: When no traffic of any kind is passing through a newly installed router interconnecting broadcast networks via an X.25 WAN, look for problems associated with the new installation. This is especially true when local area networks (LANs) previously interconnected via the WAN continue to communicate with no disruption of service.

Possible Causes and Suggested Actions

Table 7-10 outlines possible causes of connectivity problems that occur following new X.25 router installations.

Note: The process of problem isolation for DDN X.25 networks is essentially the same, but there is no static mapping capability.

Table 7-10 Causes and Actions for New Router Problems (X.25)

Possible Causes	Suggested Actions
Link is dead	<p>Step 1: Use show interfaces serial to determine whether link is down.</p> <p>Step 2: If link is down, troubleshoot serial line per CSU/DSU loopback tests and ping tests described earlier or with serial analyzer.</p>
Switch is misconfigured	<p>Step 1: Check configuration of switch; look for bad address specifications, incorrect VC parameter settings, or other configuration errors.</p> <p>Step 2: If errors are found, modify configuration and check status of the line via show interface serial output.</p>
One of the following: 1. Router is misconfigured 2. Cabling is incorrect 3. Bad router hardware	<p>Step 1: If the status line displays "Serial (n) is up, Line protocol is down," check the status of LAPB (state will probably be in CONNECT or SABMSENT).</p> <p>Step 2: If the LAPB status is <i>not</i> CONNECT, attach a serial analyzer (probably will show SABMSENT).</p> <p>Step 3: Using the serial analyzer, look for unnumbered acknowledge (UA) packets sent in reply to SABMs.</p> <p>Step 4: If UAs are not being sent, one of these problems is the likely cause.</p> <p>Step 5: Reconfigure equipment or replace as required.</p> <p>Step 6: If the status line displays "Serial (n) is up, Line protocol is up" and no connections can be made, check the router configuration with the write terminal EXEC command.</p>

Possible Causes	Suggested Actions
	<i>Step 7:</i> Check the x25 map commands and ensure that the correct addresses are specified.
	<i>Step 8:</i> Verify that the broadcast keyword is included in the x25 map command (if dynamic routing is being used in the network).
	<i>Step 9:</i> Ensure that all router configuration options match switch settings.
	<i>Step 10:</i> Modify configuration on router as needed to resume operation.

Users Cannot Connect to Resources over New Frame Relay Link

Symptom: When no traffic of any kind is passing through a newly installed router interconnecting broadcast networks via a Frame Relay WAN, look for problems associated with the new installation. This is especially true when local area networks (LANs) previously interconnected via the WAN continue to communicate with no disruption of service.

Possible Causes and Suggested Actions

Table 7-11 outlines possible causes of connectivity problems that occur following new Frame Relay router installations.

Table 7-11 Causes and Actions for New Router Problems (Frame Relay)

Possible Causes	Suggested Actions
Frame Relay switch is misconfigured (dynamic DLCI and protocol address mapping)	Step 1: Check output of show interfaces serial to determine line status and whether LMI updates have been received.
	Step 2: If LMIs have not been received, enable debug frame-relay-lmi ; look for LMI information to determine whether switch and router are sending and receiving LMI packets.
	Step 3: Check configuration of Frame Relay switch; make sure LMI is in use (dynamic mode). This is not a problem if using static mode.
Router is misconfigured (wrong keepalive setting)	Step 1: Evaluate router configuration—use write terminal EXEC command; check for LMI keepalive setting (dynamic mode).
	Step 2: Compare LMI keepalive setting with switch setting.
	Step 3: Make sure router is at least two seconds faster than switch.
Router is misconfigured (dynamic DLCI and protocol address mapping)	Step 1: Check output of show interfaces serial to determine line status.
	Step 2: Determine whether DLCI-to-protocol mapping is dynamic or static; set to correct mode if not correct.
	Step 3: If dynamic mapping is implemented and interface status is line up/protocol up, but no connections can be made, get output of show frame-relay map .
	Step 4: Determine whether any of the far end networks were learned by the local router.
	Step 5: If far end networks were learned, ping nearest interface of remote router (if protocol supports ping). Verify that you can reach that point.
	Step 6: If you <i>cannot</i> successfully ping (with line up/protocol up status), the Frame Relay network is probably misconfigured.
	Step 7: If ping works, ping through to other side of router, working out to end stations.

Possible Causes	Suggested Actions
	<p><i>Step 8:</i> Reconfigure equipment as necessary. (At the router, be sure that the remote DLCI number is mapped to the protocol address at the far end.)</p> <p><i>Step 9:</i> If no far end networks are learned with show frame-relay map, enable debug frame-relay-events and execute the appropriate show route command (protocol dependent).</p> <p><i>Step 10:</i> Determine what exchanges are occurring between router and switch and whether any routing protocol information is being picked up.</p> <p><i>Step 11:</i> Make any necessary router configuration changes to address specifications or other Frame Relay configuration entries.</p>
Router misconfigured (static Frame Relay address mapping)	<p><i>Step 1:</i> Check output of show interfaces serial to determine line status.</p> <p><i>Step 2:</i> Determine whether DLCI-to-protocol mapping is dynamic or static; set to correct mode if not correct.</p> <p><i>Step 3:</i> If static mapping is implemented with line up/protocol up interface status, but no connections can be made, get output of show frame-relay map.</p> <p><i>Step 4:</i> Status should be defined as “active”; if not, check the configurations on the switch and router, compare, and make sure they match.</p> <p><i>Step 5:</i> If show frame-relay map indicates active status, get output of appropriate show route for protocol (determine whether routing information is accumulating); proceed to access list “Possible Causes” section.</p>
Router misconfigured (bad access list implementation)	<p><i>Step 1:</i> Evaluate router access lists at both ends of connection.</p> <p><i>Step 2:</i> Make sure there are no inadvertent access denials.</p> <p><i>Step 3:</i> Modify as needed or remove to test; rework to allow appropriate access.</p>
Cabling problem	<p><i>Step 1:</i> Inspect line/protocol status and check cabling per serial diagnostics discussions earlier in this chapter.</p> <p><i>Step 2:</i> Replace any incorrectly configured or failed cables.</p>
Dead hardware	<p><i>Step 1:</i> Inspect line/protocol status and perform loopback and ping tests as described earlier in this chapter to isolate specific problem equipment.</p> <p><i>Step 2:</i> Replace hardware as necessary.</p>

Users Cannot Connect to Resources over New SMDS Link

Symptom: When no traffic of any kind is passing through a newly installed router interconnecting broadcast networks via an SMDS WAN link, look for problems associated with the new installation. This is especially true when local area networks (LANs) previously interconnected via the WAN continue to communicate with no disruption of service.

Possible Causes and Suggested Actions

Table 7-12 outlines possible causes of connectivity problems that occur following new SMDS router installations.

If you are having difficulty establishing connections over an SMDS cloud, obtain the following information as a preliminary step before beginning the problem isolation process:

- Use the **show arp** command to determine whether any SMDS devices were detected on the switch.
- Use the **debug serial-interface** command to determine whether packets are being sent and received.
- Use the **debug serial-packet** command to obtain the entire SMDS header and payload data when SMDS packets are transmitted or received on an interface.
- Use the **debug arp** command to determine what other SMDS devices are being detected on the switch.

Table 7-12 Causes and Actions for New Router Problems (SMDS)

Possible Causes	Suggested Actions
SMDS switch is misconfigured	<i>Step 1:</i> Check router and switch configurations for address mismatch. <i>Step 2:</i> Make sure SMDS switch is configured for multicast or static mapping (depending on intended network setup).
Router misconfigured (general SMDS)	<i>Step 1:</i> Evaluate router configuration—using write terminal EXEC command. <i>Step 2:</i> Compare router configuration with switch requirements. Examples: bad address specification, wrong mode (multicast vs. static), smds encapsulation not included in configuration. <i>Step 3:</i> Modify configuration if necessary to make router match SMDS network requirement.
Misconfigured SMDS interface or multicast addresses	<i>Step 1:</i> Use the show arp command to determine what other devices, if any, have been detected on the switch. <i>Step 2:</i> If none are being detected, check to make sure that the interface SMDS address specified in the smds address configuration command matches the address of the attached switch. <i>Step 3:</i> Check to make sure that the SMDS multicast addresses specified in the smds multicast configuration command match the addresses configured on the switch.

Possible Causes	Suggested Actions
	<p><i>Step 4:</i> Make sure that ARP is enabled with the smds enable-arp command so that higher layers learn about the router.</p> <p><i>Step 5:</i> Check the router's configuration of static maps. Make sure that static maps are configured for all nonlearning protocols to allow the SMDS software to translate a destination address into a proper SMDS address for outgoing packets.</p> <p><i>Step 6:</i> If SMDS data is still not being received, even if packets are being sent, check the cable/SDSU/AU/Switch connections for proper physical connectivity.</p> <p><i>Step 7:</i> If the physical connections are operational, and packets still are not being received, check the SDSU configuration.</p>
Router misconfigured (static SMDS address mapping)	<p><i>Step 1:</i> Check output of show interfaces serial to determine line status.</p> <p><i>Step 2:</i> Using the show smds map command, determine whether applicable mode is multicast or static; configure correct mode if not correct.</p> <p>All network protocols, with the exception of IP and ISO CLNS, require static mapping from the protocol address to SMDS addresses.</p> <p><i>Step 3:</i> If IP or ISO CLNS are being routed, check the multicast group specification. Make any necessary address changes.</p> <p><i>Step 4:</i> If static mapping is implemented with line up/protocol up interface status, but no connections can be made, enable debug serial-interface.</p> <p><i>Step 5:</i> Based on the debug output, determine whether the correct destination address is being used.</p> <p><i>Step 6:</i> Make configuration changes as necessary to mapping, mode, or encapsulation specification.</p>
Router misconfigured (bad access list implementation)	<p><i>Step 1:</i> Evaluate router access lists at both ends of connection.</p> <p><i>Step 2:</i> Make sure there are no inadvertent access denials.</p> <p><i>Step 3:</i> Modify as needed or remove to test; rework to allow appropriate access.</p>
Cabling problem	<p><i>Step 1:</i> Inspect line/protocol status and check cabling per serial diagnostics discussions earlier in this chapter.</p> <p><i>Step 2:</i> Replace any incorrectly configured or failed cables.</p>
Dead hardware	<p><i>Step 1:</i> Inspect line/protocol status and perform loopback and ping tests as described earlier in this chapter to isolate specific problem equipment.</p> <p><i>Step 2:</i> Replace hardware as necessary</p>

Some Users Cannot Connect to Resources over WAN

Symptom: When some users or applications are able reach resources over a serial/WAN link through a router, while others cannot, the problem usually can be attributed to a configuration error.

Possible Causes and Suggested Actions

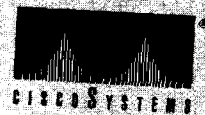
Table 7-13 outlines possible causes of selective connectivity problems in serial and WAN interconnections.

Table 7-13 Causes and Actions for Selective Connectivity Problems

Possible Causes	Suggested Actions
Bad access list specification	Step 1: Check configuration using write terminal . Look for any access list specifications.
	Step 2: If found, carefully compare access list with requirements. Be sure that no implicit access deny (or explicit access deny) is blocking required access.
	Step 3: Modify access lists as needed.
Host configuration not set up to send ARPs	Step 1: Check host configuration.
	Step 2: Modify host configuration to send ARPs.
Host configuration points at wrong router	Step 1: Check host configuration for gateway of last resort specification.
	Step 2: Modify host configuration for gateway of last resort.
Discontinuous subnet addressing (IP)	Step 1: Check network configuration for discontinuous network address space assignment.
	Step 2: If found, use secondary IP address to accommodate physical discontinuity.

Part 2

Profile Handling of Insurance



Chapter 8

Performance Problem Scenarios

8

Performance Scenarios: Overview and List 8-2

Performance Problems in Novell IPX Internet After Bandwidth Upgrade 8-3

- Symptoms 8-3
- Environment Description 8-3
- Diagnosing and Isolating Problem Causes 8-4
- Problem Solution Summary 8-4

Performance Problems in Novell IPX Internet After Switch to Routing 8-5

- Symptoms 8-5
- Environment Description 8-5
- Diagnosing and Isolating Problem Causes 8-6
- Problem Solution Summary 8-6

Slow Novell IPX Performance over Router Connecting 16-Mbps Rings 8-7

- Symptoms 8-7
- Environment Description 8-7
- Diagnosing and Isolating Problem Causes 8-8
- Problem Solution Summary 8-8

Slow Novell Performance over Ethernet Backbone 8-9

- Symptoms 8-9
- Environment Description 8-9
- Diagnosing and Isolating Problem Causes 8-10
 - Identifying Congestion as the Problem 8-10
- Problem Solution Summary 8-10

Slow Novell Performance over Matching Parallel Links 8-13

- Symptoms 8-13
- Environment Description 8-13
- Diagnosing and Isolating Problem Causes 8-14
 - Isolating and Resolving Uneven Load Problem 8-14
- Problem Solution Summary 8-14

Slow Novell Performance over Unequal Parallel Links 8-15

Symptoms 8-15

Environment Description 8-15

Diagnosing and Isolating Problem Causes 8-16

 Isolating and Resolving Uneven Load Problem 8-16

Problem Solution Summary 8-16

Poor Performance over TCP/IP Serial Network 8-17

Symptoms 8-17

Environment Description 8-17

Diagnosing and Isolating Problem Causes 8-18

Problem Resolution Process 8-18

 Isolating Problem Location 8-18

Problem Solution Summary 8-20

Slow Host Response over 56-Kbps HDLC Link 8-21

Symptoms 8-21

Environment Description 8-21

Diagnosing and Isolating Problem Causes 8-22

Problem Resolution Process 8-22

 Isolating Serial Hardware and Media Problems 8-22

Problem Solution Summary 8-27

Chapter 8

Performance Problem Scenarios

8

This chapter presents problem-solving *scenarios* focusing on identifying, isolating, and solving problems that impede throughput performance in internetworks.

These example problem-solving scenarios address specific situations and illustrate the process of problem isolation and resolution. The scenarios provided here span different protocols, media, and problem types. The objective is to illustrate a problem-solving method based on the problem-solving model defined in Chapter 1, “Troubleshooting Overview.” The scenarios that follow focus specifically on situations in which traffic is getting to its intended destination, but network users complain about slow host response, connections dropping, or sporadic resource availability.

Each scenario includes the following components:

- Symptom statement
- Internetworking environment description
- Problem isolation discussion and process
- Solution summary

Chapter 9, “Troubleshooting Internet Performance,” presents a series of *symptom modules* that provide snapshots of common symptoms, possible causes, and suggested actions for the protocols and technologies addressed in this publication.

An overview of scenarios and symptom modules is provided in “How to Use this Publication” in Chapter 1, “Troubleshooting Overview.”



Caution: Throughout this and other chapters, the use of **debug** commands is suggested for obtaining information about network traffic and router status. Use these commands with great care. In general, it is recommended that these commands only be used under the direction of your router technical support representative when troubleshooting specific problems. Enabling debugging can disrupt operation of the router when internets are experiencing high load conditions. When you finish using a **debug** command, remember to disable it with its specific **undebug** command or with the **undebug all** command.

Performance Scenarios: Overview and List

In general, performance slowdowns are considered lower-priority problems than reachability issues. However, poorly performing internetworks can degrade organizational productivity and often can effectively halt operation of network applications if communications degenerate enough. Performance problems can manifest themselves in many ways. Slow host response, dropped connections, and high error counts all suggest that network performance is not optimal. Unfortunately, the actual sources of performance problems are often difficult to detect.

This chapter presents a series of situational discussions, including the application of various diagnostic tools. Every possible scenario cannot be covered. Indeed, the scenarios included here only scratch the surface of possible situations. However, certain common themes typically tie all connectivity problems together. This chapter is provided in an effort to illustrate the use of troubleshooting tools and techniques to identify those common themes.

The following problem-solving scenarios are presented in this chapter:

- **Novell Performance Scenarios**—Seven short performance-related case examples illustrate typical configuration and network design problems that can lead to poor performance in Novell IPX internets.
- **Poor Performance over TCP/IP WAN**—Focuses on performance in a TCP/IP internetwork featuring parallel serial links joining two geographically separated locations via Cisco routers.
- **Serial Link Performance Scenario**—Persistent complaints by users about poor host response over a 56-Kbps HDLC link.

Performance Problems in Novell IPX Internet After Bandwidth Upgrade

The following case illustrates a situation in which performance degrades significantly after a Novell IPX internet is "upgraded" from a 2400-baud link over a phone line to a 9600-baud synchronous serial line.

Symptoms

Server responsiveness noticeably slows following an upgrade from a 2400-baud, direct dial-up interconnection between a client and a server to router-based link over a 9600-baud synchronous serial line.

Environment Description

Figure 8-1 illustrates a map of the internetwork change discussed in this case. The following list summarizes relevant elements of the environment:

- Initial communication between Client-A and Server-A is acceptable when provided over a direct dial-up link.
- In order to share resources, Client-A is attached to an Ethernet, and the dial-up access is replaced by 9600-baud synchronous serial line separated by two routers.
- The LANs are IEEE 802.3 Ethernets.
- Novell IPX is the only protocol being routed over the point-to-point link

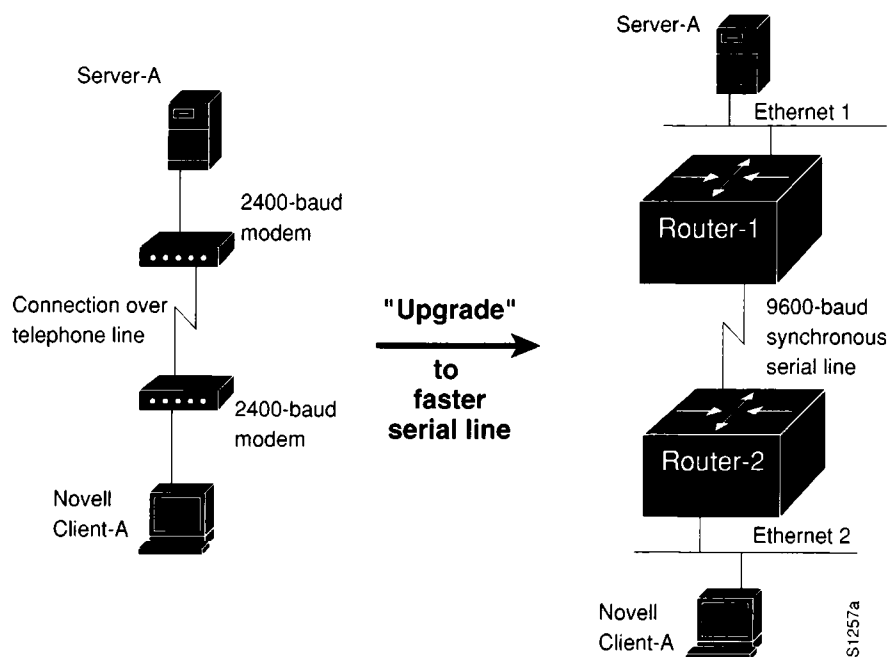


Figure 8-1 Upgrade from Dial-Up Link to 9600-Baud Connection

Diagnosing and Isolating Problem Causes

Given the situation, insufficient bandwidth is the best candidate for poor server responsiveness.

As illustrated in Figure 8-1, the “upgraded” environment will be inherently slower than the original direct, dial-up interconnection.

In the original configuration, Server-A communicates with Client-A without any encapsulation. Although the modems do attach a header to each transmission, information exchanged between Server-A and Client-A is essentially all data and specifically varies in size depending on the kind of communication occurring.

In the “upgraded” configuration, the Ethernet segments to which Server-A and Client-A are attached require a minimum packet size of 60 bytes (which includes a six-byte destination address, a six-byte source address, a two-byte type or length field, and data). The overhead associated with Ethernet encapsulation (for packets smaller than 60 bytes) can easily overwhelm the 9600-baud line.

Problem Solution Summary

One possible solution is to disable fast switching. When fast switching is disabled, the router uses the network layer packet size instead of the link layer packet size (as in fast switching). In addition, more buffering is available to handle peak loads when fast switching is disabled.

However, with such a narrow serial pipe, adding bandwidth is your best option. The question is, how much bandwidth does the application require?

The amount of additional bandwidth required will vary depending on your situation. Certainly, if multiple clients are trying to access multiple servers, converting the 9600-baud line to a 56-Kbps line would be reasonable.

Performance Problems in Novell IPX Internet After Switch to Routing

The following case illustrates a situation in which performance degrades significantly after a bridged Novell IPX internet is converted to routing.

Symptoms

Server responsiveness slows by an approximate factor of four after Novell IPX routing is implemented in place of bridging.

Environment Description

Figure 8-2 illustrates a map of the internetwork change discussed in this case.

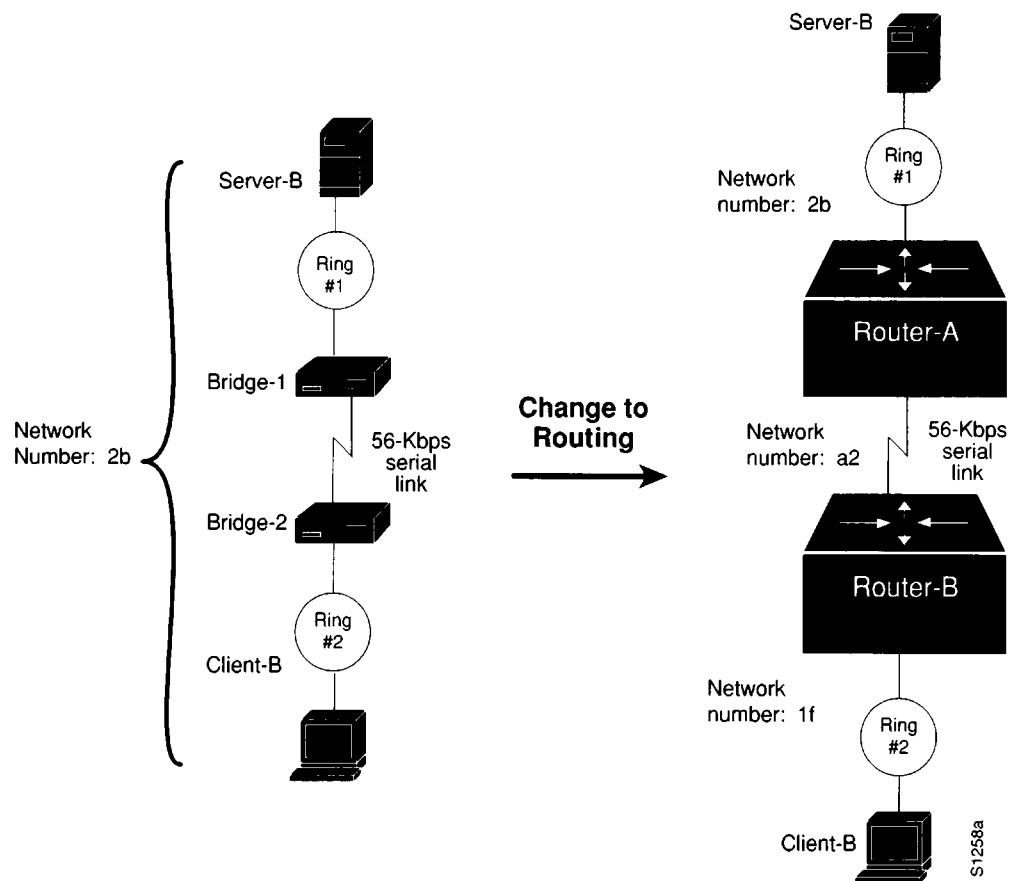


Figure 8-2 Novell IPX Interconnection Converted from Bridging to Routing

The following list summarizes relevant elements of the environment:

- Previously, communication between Client-B and Server-B is provided via two remote bridges over a 56-Kbps link.
- In order to ensure a more manageable interconnection, Bridge-1 and Bridge-2 are replaced with routers (Router-1 and Router-2) over the same 56-Kbps link.
- The LANs are IEEE 802.5 Token Rings.
- Novell IPX is the only protocol being routed over the point-to-point link.

Diagnosing and Isolating Problem Causes

Given the situation, the maximum packet size limitation associated with standard NetWare in a routed environment is the best candidate for poor server responsiveness.

In a bridged environment, Server-B allows transmission of the maximum packet size associated with the media in the internetwork (1130 bytes for Ethernet, 4202 bytes for 4-Mbps Token Ring and 16-Mbps Token Ring).

However, in a router-based internet, standard Novell servers allow for a maximum packet size of 512 bytes, regardless of media. Packet routing defaults to this smallest common size whenever multiple network numbers are detected. In addition, prior to Software Release 8.3 (3), Cisco routers did not support Novell's Large Internet Packet Exchange NetWare-loadable module (*LIPX.NLM*) on Token Ring. This module was previously referred to as *BIGPACK.NLM*.

Problem Solution Summary

There are three actions that can help improve performance between Server-B and Client-B in this router-based internet:

1. Upgrade the routers to Software Release 8.3(3) or higher for support of the *LIPX.NLM* Novell NetWare-loadable module.
2. Implement *LIPX.NLM* on the server to accommodate the transmission of packets of any size requested by clients.



Caution: Be sure that all media and transport protocols are accounted for when implementing *LIPX.NLM*. If any segment does not support the larger packets, connectivity can be disrupted throughout the internet.

3. Implement Novell's *PBURST.NLM* NetWare-loadable module at the server and *BNETX.COM* at clients to support "burst mode." This software implements a "windowing" capability, allowing for larger individual units of data transfer.

Slow Novell IPX Performance over Router Connecting 16-Mbps Rings

The following case illustrates a situation in which performance over a router interconnecting two 16-Mbps Token Rings is slower than a comparable interconnection of two Ethernet segments.

Symptoms

Server responsiveness is slow over the router separating two 16-Mbps rings.

Environment Description

Figure 8-3 illustrates a map of the relevant elements of this environment:

- Router-1 separates two Token Ring segments.
- Client-C (Ring #2) is accessing services on Server-C (Ring #1).
- The LANs are 16-Mbps Token Rings.
- Novell IPX is the only protocol being routed.

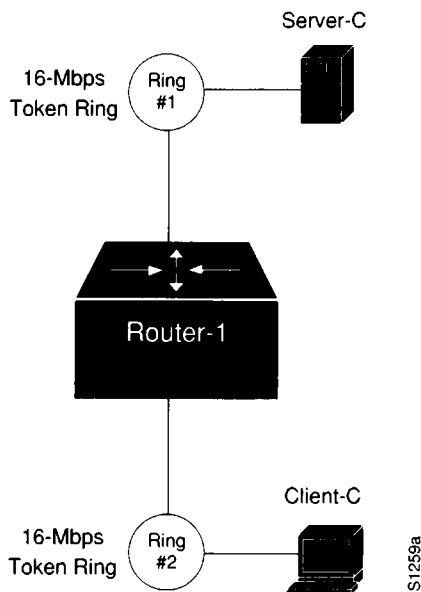


Figure 8-3 Novell IPX Interconnection over Router Joining 16-Mbps Rings

Diagnosing and Isolating Problem Causes

Given the situation, the router is probably implementing system software that predates Software Release 9.1. This is the best candidate for poor server responsiveness.

Prior to Software Release 9.1, Novell IPX routing over Token Ring was slow switched. As of Software Release 9.1, the routers support fast switching of Novell IPX.

Prior to Software Release 8.3(3), Cisco routers did not support Novell's Large Internet Packet Exchange NetWare-loadable module (*LIPX.NLM*) for Token Ring. This module was previously referred to as *BIGPACK.NLM*.

Problem Solution Summary

Use the **show version** command to determine the software revision level.

Assuming that the software version predates Software Release 9.1, the solution to this situation is as follows:

1. Upgrade the router to Software Release 9.1, thereby enabling fast switching of Novell IPX traffic by default over Token Ring interfaces.
2. Since Software Release 8.3(3) or higher can support *LIPX.NLM*, implement *LIPX.NLM* on the server, allowing for transmission of packets of any size requested by clients.



Caution: Be sure that all media and transport protocols are accounted for when implementing *LIPX.NLM*. If any segment does not support the larger packets, connectivity can be disrupted throughout the internet.

3. Implement Novell's *PBURST.NLM* NetWare-loadable module at the server and *BNETX.COM* at clients to support "burst mode." This software implements a "windowing" capability, allowing for larger individual units of data transfer.

Slow Novell Performance over Ethernet Backbone

The following case illustrates a situation in which performance is extremely slow over an Ethernet backbone separating two routers.

Symptoms

Slow server response among multiple Ethernet segments separated by two routers and an Ethernet backbone.

Environment Description

Figure 8-4 illustrates a map of the relevant elements of this environment:

- Router-A and Router-B interconnect multiple Ethernets via an Ethernet backbone.
- All LANs are IEEE 802.3 Ethernets.
- Novell IPX is the only protocol being routed.

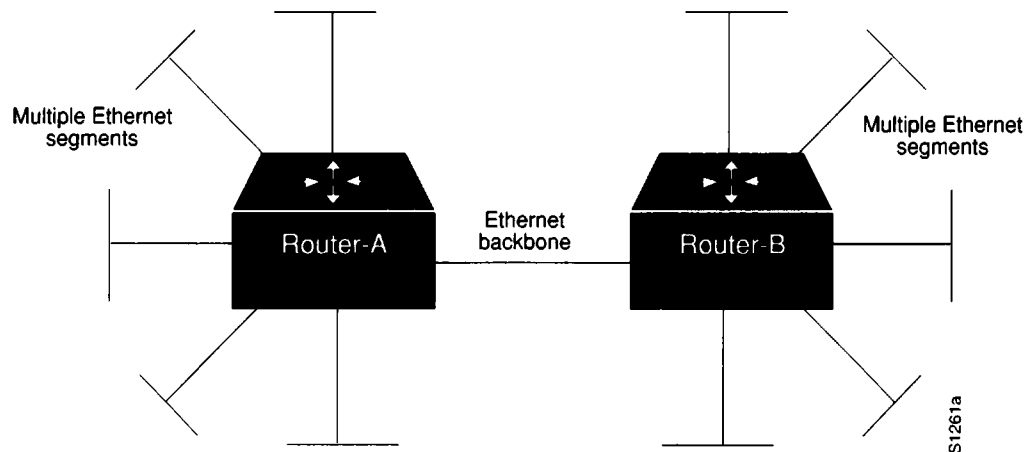


Figure 8-4 Novell IPX Router Joining Ethernet and Token Ring

Diagnosing and Isolating Problem Causes

Given the situation, congestion is the best candidate for poor performance over the backbone.

The following discussion briefly outlines the process of problem isolation for the situation discussed.

Identifying Congestion as the Problem

In general, you can use the following two methods to determine whether there is a congestion problem over the backbone:

- Step 1:** Use the **show interfaces** command and examine the display output for relative load, high and increasing levels of input errors, and drops.
- Step 2:** Attach a network analyzer onto the backbone and look for high levels of collisions and bandwidth utilization in excess of 30 percent.

Refer to “Slow Host Response over 56-Kbps HDLC Link” earlier in this chapter for information about general troubleshooting of performance problems in a routed internet. Also, refer to Chapter 7, “Troubleshooting WAN Connectivity,” and Chapter 9, “Troubleshooting Internet Performance,” for more information about diagnosing congestion problems.

If you do determine that congestion over the Ethernet backbone is high, the only real option is to increase bandwidth. You can do this by either adding additional Ethernet segments or by replacing the Ethernet backbone with a faster media, such as FDDI.

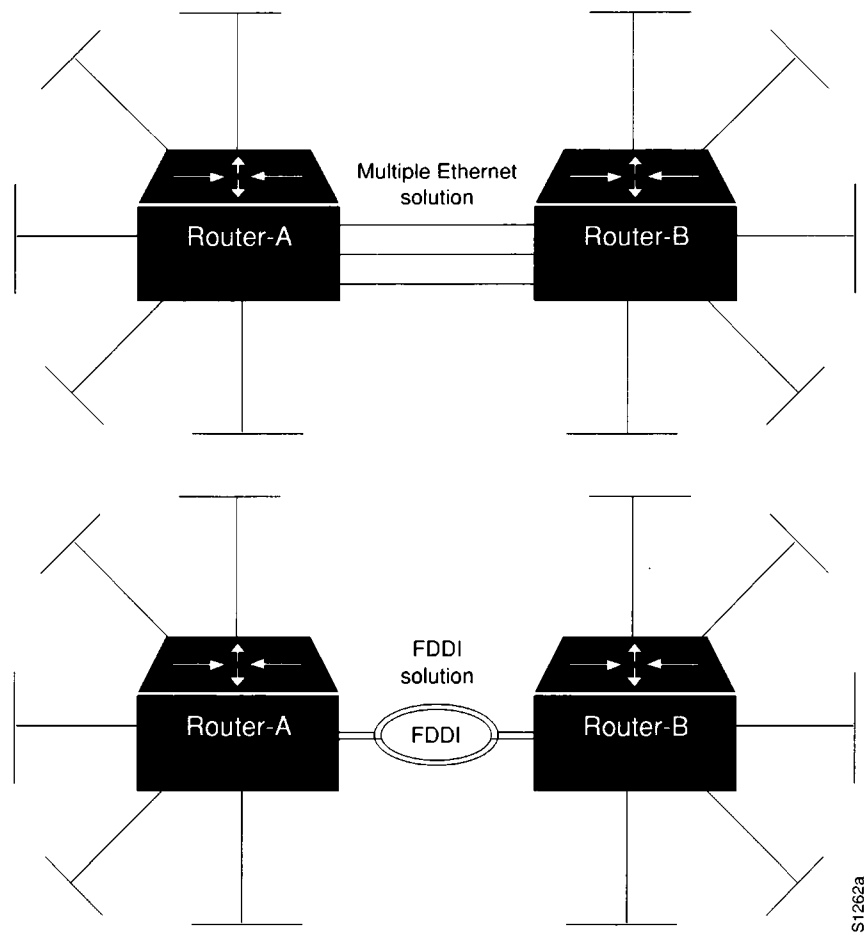
Problem Solution Summary

This scenario focused on improving performance over a backbone segmenting multiple Ethernets by increasing bandwidth using one of two options:

- Replacing the single Ethernet backbone with multiple Ethernet segments.
- Replacing the single Ethernet backbone with an FDDI backbone.

Figure 8-5 illustrates these options.

Note: If you adopt a multiple Ethernet option, remember to implement the **novell maximum-paths** *paths* global configuration command. Refer to the subsequent scenario entitled “Slow Novell Performance over Matching Parallel Links” for more information about this requirement. Also note that each segment must have its own network address. Refer to Chapter 5, “Troubleshooting Novell Connectivity,” for more discussions about duplicate network number problems.



S1262a

Figure 8-5 Alternative Solutions to Ethernet Backbone Bottleneck

Slow Novell Performance over Matching Parallel Links

The following case illustrates a situation in which performance is less than optimal over parallel T1 links joining two routers.

Symptoms

One line appears to be heavily loaded, while the other is either idling or indicates very low load. Users complain of slow response and intermittent connection drops.

Environment Description

Figure 8-6 illustrates a map of the relevant elements of this environment:

- Router-X and Router-Y interconnect two sites over parallel T1 lines running at 1.544 Mbps.
- Client-E needs to access Server-E on the other side of the serial interconnections.
- All LANs are IEEE 802.3 Ethernets.
- Novell IPX is the only protocol being routed.

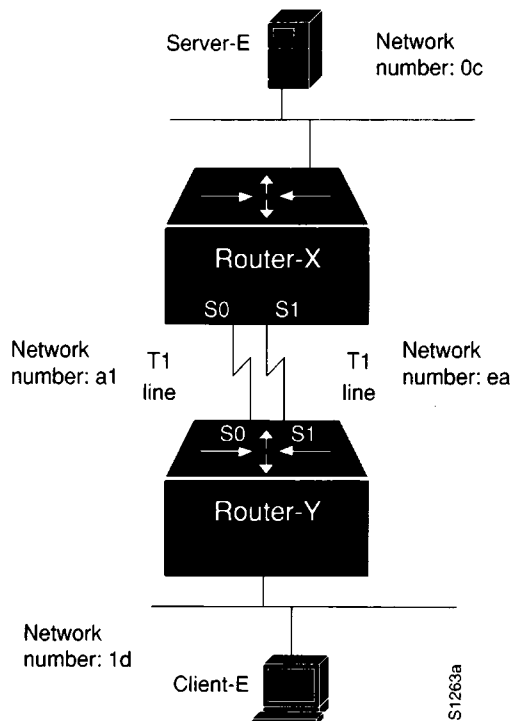


Figure 8-6 Router Joining Novell IPX Networks over Parallel T1 Lines

Diagnosing and Isolating Problem Causes

Given the situation, the router probably is keeping only one routing table entry per target network. This is very likely to cause poor performance over the parallel serial lines. In the worst case, traffic is only routed through one line, while the second line is idle.

Isolating and Resolving Uneven Load Problem

In general, you can use the following method to determine whether traffic is being unevenly distributed between the parallel lines:

- Step 1:** Use the **show interfaces** command and examine the displayed load for each interface. Also examine the number of input and output drops and the five-minute output and input error counts. Record the observed values.
- Step 2:** Use the **clear counters** command and continue to monitor changes in the counters over time with the **show interfaces** command.
- Step 3:** Look for values that are substantially uneven, (for example serial interface S0 indicates 300,000 packets total input, while serial interface S1 indicates only 1,000).
- Step 4:** If you determine that traffic is being unevenly distributed over the serial links, use the **novell maximum-paths paths** command to set the number of multiple paths the router will use when transmitting traffic to any particular destination. Instead of keeping only one routing table entry, the router will now remember up to the number of paths specified when determining how to route traffic. In essence, this forces load balancing over the two lines when *paths* is set to **2**.

Note: This problem is the same for any parallel media. The suggested solution would be the same for parallel FDDI, Ethernet, or Token Ring links, as well as for parallel serial interconnections.

Problem Solution Summary

This scenario focused on improving performance over parallel links. The recommended solution here is to implement the **novell maximum-paths paths** global configuration command with a *paths* specification of **2**.

Slow Novell Performance over Unequal Parallel Links

The following case illustrates a situation in which performance is slow over parallel links of differing speeds that join two routers.

Symptoms

One line appears to be heavily loaded, while the other is either idling or indicates very low load. Users complain of slow response and intermittent connection drops.

Environment Description

Figure 8-7 illustrates a map of the relevant elements of this environment:

- Router-R and Router-S interconnect two sites over parallel lines with one running at full T1 speed, 1.544 Mbps, and the other running at 9.6 Kbps.
- The **novell maximum-paths** command is enabled on both routers.
- Client-N needs to access Server-F on the other side of the serial interconnections.
- All LANs are IEEE 802.3 Ethernets.
- Novell IPX is the only protocol being routed.

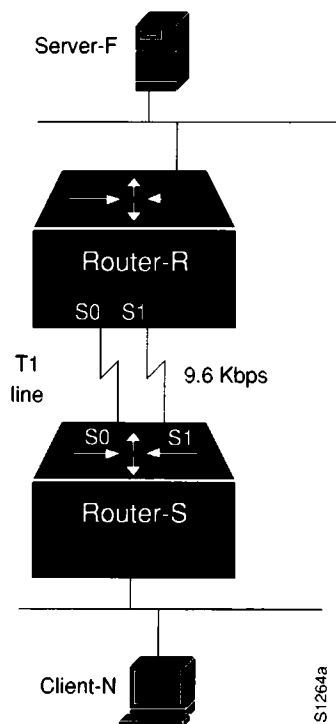


Figure 8-7 Router Joining Novell IPX Networks over Uneven Parallel Lines

Diagnosing and Isolating Problem Causes

Because Novell's RIP routing protocol does not account for line speed, load cannot be balanced effectively between these two links. Given the situation, this load-balancing problem is probably causing poor performance over the parallel serial lines.

As in the prior case, it is quite possible that traffic will only be routed through one line, while the second line is idle. And, since RIP does not consider line speed, the 9.6-Kbps line could be completely overwhelmed, while the T1 line is relatively quiet.

Isolating and Resolving Uneven Load Problem

In general, you can use the same method to determine whether traffic is being unevenly distributed between the parallel lines as discussed in the prior case:

- Step 1:** Use the **show interfaces** command and examine the displayed load for each interface. Also examine the number of input and output drops and the five-minute output and input error counts. Record the observed values.
- Step 2:** Use the **clear counters** command and continue to monitor changes in the counters over time with the **show interfaces** command.
- Step 3:** Look for values that are substantially uneven, (for example Serial0 indicates 300,000 packets total input, while Serial1 indicates only 1000).
- Step 4:** If you determine that traffic is being unevenly distributed over the serial links, and the **novell maximum-paths paths** command is already implemented, your only solution is to make the speed on both lines match or eliminate the slow-speed line altogether.

Note: This problem is the same for any differing parallel media. The suggested solution would be the same for unevenly matched FDDI, Ethernet, or Token Ring links, as well as for uneven parallel serial interconnections.

Problem Solution Summary

This scenario focused on improving performance over uneven parallel links. The recommended solution here is to force the speed of the parallel links to match or to eliminate the slow-speed link.

Poor Performance over TCP/IP Serial Network

The problem scenario that follows focuses on performance in a TCP/IP internetwork featuring parallel serial links joining two geographically separated locations via Cisco routers. The analysis focuses on isolating problems and then considering options for relieving congestion.

Symptoms

R&D users at Remote-Lab complain of poor host response and slow performance when connecting to hosts at the Main-Campus. In addition, during certain times of the day, large files are being transferred over the serial network. At these times, traffic becomes especially slow, but does not stop.

Environment Description

Figure 8-8 illustrates a map of the internetwork discussed in this case. The following list summarizes relevant elements of the environment:

- A remote research lab (Remote-Lab) is linked to a campus network (Main-Campus) over two parallel 56-Kbps HDLC lines (Serial-X and Serial-Z).
- Two routers (Router-Main and Router-Lab) join the two sites. The routers are attached to CSU/DSUs via V.35 cables.
- The LANs are IEEE 802.3 Ethernets.
- UNIX workstations are used at the Remote-Lab; traffic to the Main-Campus consists of FTP, Telnet, and Mail.
- TCP/IP is being routed over the point-to-point links.

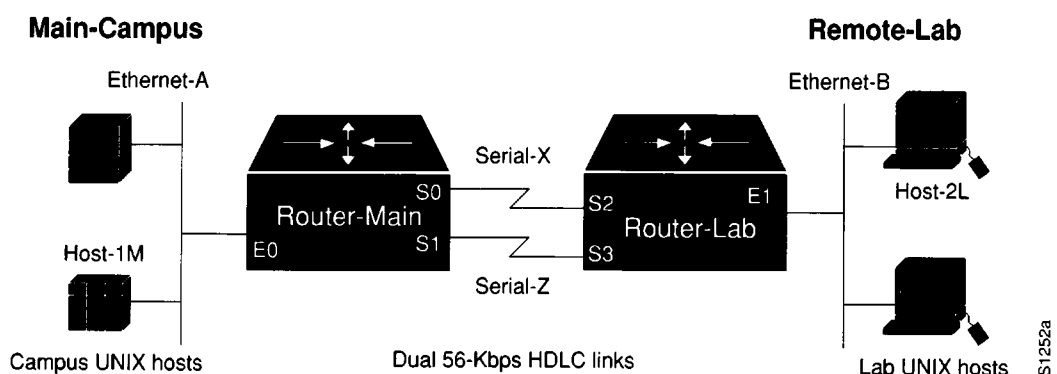


Figure 8-8 Dual 56-Kbps Serial Link TCP/IP Internet Scenario Map

Diagnosing and Isolating Problem Causes

The first thing to do is to identify all likely or possible causes; the second step is to eliminate each one. Given the situation, the following problems are the best candidates for interconnection failure:

- Bad Ethernet or serial line
- Congestion

Problem Resolution Process

The following discussion works through the process of problem isolation, then suggests possible solutions.

Isolating Problem Location

The following procedure illustrates the process of investigating potential hardware problems.

- Step 1:** The first thing to do is to determine the condition of the serial lines. A preliminary check can be done using the **show interfaces EXEC** command. Figure 8-9 illustrates a typical display that would be returned by the system if the interfaces are minimally operational and the system can communicate with them.

```
Serial 0 is up, line protocol is up
Hardware is MCI Serial
MTU 1500 bytes, BW 56 Kbit, DLY 20000 usec, rely 255/255, load 1/255
Encapsulation HDLC, loopback not set, keepalive set (10 sec)
Last input 00:00:04, output 00:00:08, output hang never
Last clearing of "show interface" counters never
Output queue 0/40, 0 drops; input queue 0/75, 40 drops
Five minute input rate 0 bits/sec, 0 packets/sec
Five minute output rate 0 bits/sec, 0 packets/sec
  240 packets input, 15768 bytes, 0 no buffer
    Received 0 broadcasts, 0 runts, 0 giants
  2 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
 174 packets output, 11432 bytes, 0 underruns
   0 output errors, 0 collisions, 40 interface resets, 0 restarts
   0 carrier transitions
```

Figure 8-9 Display Output of Show Interfaces Command

Look for input errors and output drops being high. These would suggest that the serial line is being overutilized.

- Step 2:** Assume that the serial line is determined to be *basically* functional. That is, you see that the router reports “Serial 0 is up, line protocol is up.” Now, use an extended **ping** test to isolate where traffic is being slowed. Look for drops, failures, and timeouts. Figure 8-10 illustrates an example of an extended **ping** test where failures are detected.

```

dingus#ping
Protocol [ip]:
Target IP address: 131.108.25.75
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address: dingus
Translating "DINGUS"...domain server (255.255.255.255) [OK]

Type of service [0]: ftp
Set DF bit in IP header? [no]: n
Data pattern [0xABCD]: ffff
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 131.108.25.75, timeout is 2 seconds:
Packet has data pattern 0xFFFF
.....
Success rate is 0 percent

```

Figure 8-10 Example Ping Command Specification and Output

Step 3: Ping various nodes in the path, starting with the router closest to the remote hosts, looking for the point where drops start occurring.

For instance, **ping** from Router-Lab to Host-2L. If pings are successful, Ethernet-B can be eliminated as the source of congestion problems.

Next, **ping** from Router-Main to Host-1M. If pings are successful, Ethernet-A can be eliminated as the source of congestion problems.

Step 4: If these tests indicate no problems, ping between the routers. First, ping from Router-Main to the IP address associated with interface Ethernet1 on Router-Lab. Next, ping each of the serial interfaces on Router-Lab. If you find any ping failure on the serial lines, refer to serial debugging as discussed in Chapter 7, "Troubleshooting WAN Connectivity," and to the additional information provided in Chapter 1, "Troubleshooting Overview."

Step 5: If you determine that the problem is indeed one of congestion because of bandwidth overutilization, you must decide whether it is more effective to add bandwidth (in the form of another serial circuit) or to make an adjustment in the router configuration.

Step 6: If you see load values of about 50 percent, input drops, and output drops, you might consider implementing priority queuing to force Telnet to be given higher precedence over other packet types. This helps ensure reasonable connection service to users, even during periods when file transfers are taking place. Figure 8-11 illustrates a configuration for Router-Lab that establishes priority queuing and assigns port 23 (Telnet) a higher priority than other TCP/IP protocols, such as mail (port 25).

```
!  
priority-list 4 protocol ip medium tcp 23  
!  
interface serial 2  
ip-address 131.108.155.21 255.255.255.0  
priority-group 4  
!  
interface serial 3  
ip-address 131.108.156.22 255.255.255.0  
priority-group 4
```

Figure 8-11 Configuration Showing Priority Queuing Specification

Note: One reason why Telnet traffic can be bumped from the buffer queues is the tendency of the larger FTP packet types to collect in the router's buffers. When FTP traffic is high, the smaller Telnet packets are squeezed out of the input or output queues, resulting in retransmissions, session timeouts, and generally slower connection performance. By using priority queuing, marginal cases can be relieved.

Step 7: If you are seeing load of close to 90 percent, as well as input errors and output drops, priority queuing is not likely to do any good. With consistently high congestion, your best solution is additional or faster serial links.

Problem Solution Summary

This scenario focused on the following topics in resolving performance problems in TCP/IP internetworks:

- Isolating problem nodes and eliminating potential problems using extended **ping** tests.
- Determining when to tweak your configuration with priority queuing and when to add bandwidth.
- Specifying priority queuing to force the router to give a specific TCP/IP socket a higher priority than other protocols.

Slow Host Response over 56-Kbps HDLC Link

When designing and implementing internetworks, it is important to factor in any potential changes and expectations of growth. This is especially important when certain network elements are at risk of becoming bottlenecks—such as point-to-point serial links. Bandwidth that appears to be sufficient today may be inadequate in a year. And the budget may not exist to add another drop or replace the existing service.

The problem scenario that follows explores a situation in which performance over a serial link is not meeting user requirements. In this case, the analysis focuses on isolating the problem and then using certain router configuration capabilities to ease throughput bottlenecks.

Symptoms

Users at Remote-Site complain of consistently degraded performance when connecting to hosts at the Home-Office. Performance previously was acceptable, but now slows substantially during peak use periods.

Environment Description

Figure 8-12 illustrates a map of the internetwork discussed in this case. The following list summarizes relevant elements of the environment:

- A single remote sales office (Remote-Site) is linked to the corporate network (Home-Office) over a 56-Kbps HDLC line.
- Two routers (Router-Home and Router-Far) join the two sites via a 56-Kbps link. The local router attachment is to a CSU/DSU using a V.35 cable.
- The LANs implemented are IEEE 802.3 Ethernets.
- DEC workstations and various ASCII terminals are used at the Remote-Site; traffic to the Home-Office consists of file transfers, virtual terminal connections, and electronic mail.
- Native DECnet is routed over the point-to-point link, while DEC's Local Area Transport (LAT) protocol is the sole protocol being bridged.
- In this situation, the observed level of traffic is very high on the serial link. Spikes of 80 to 90 percent of bandwidth are commonly detected.

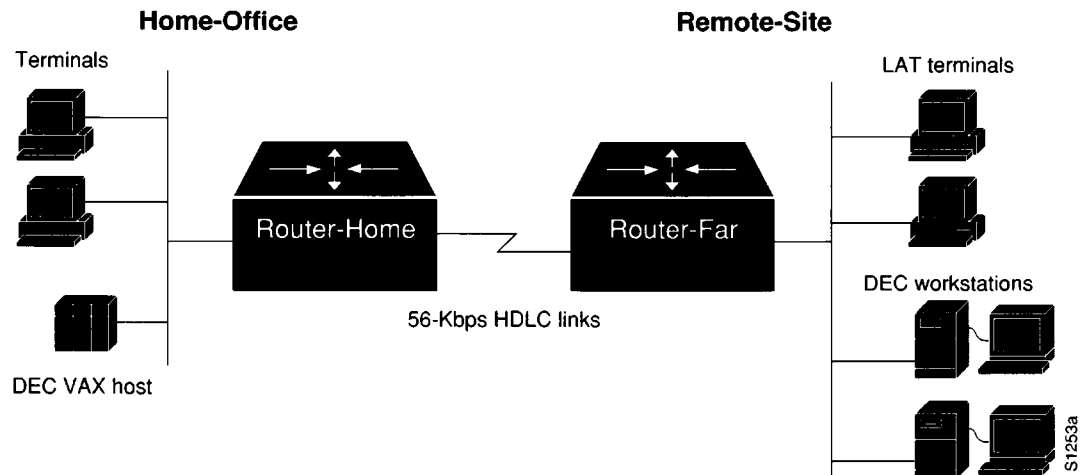


Figure 8-12 56-Kbps Point-to-Point Performance Problem Scenario Map

Diagnosing and Isolating Problem Causes

Given the situation, the following problems are the best candidates for interconnection failure:

- Bad serial line
- Overutilized serial line
- Interface card out of buffers
- Misconfigured hosts

The following discussion works through the process of problem isolation. Upon inspection, it appears that the serial interface is being overutilized (in other words, the 56-Kbps bandwidth is no longer sufficient). The question is: What can be done, short of installing another drop?

Problem Resolution Process

The discussion that follows isolates the problem, then suggests a configuration-based solution.

Isolating Serial Hardware and Media Problems

The following procedure illustrates the process of investigating potential hardware problems.

Step 1: The first thing to do is to determine the condition of the serial line. A preliminary check can be done using the **show interfaces EXEC** command. Figure 8-13 illustrates a typical display that the system returns when the interfaces are minimally operational and the system can communicate with them.

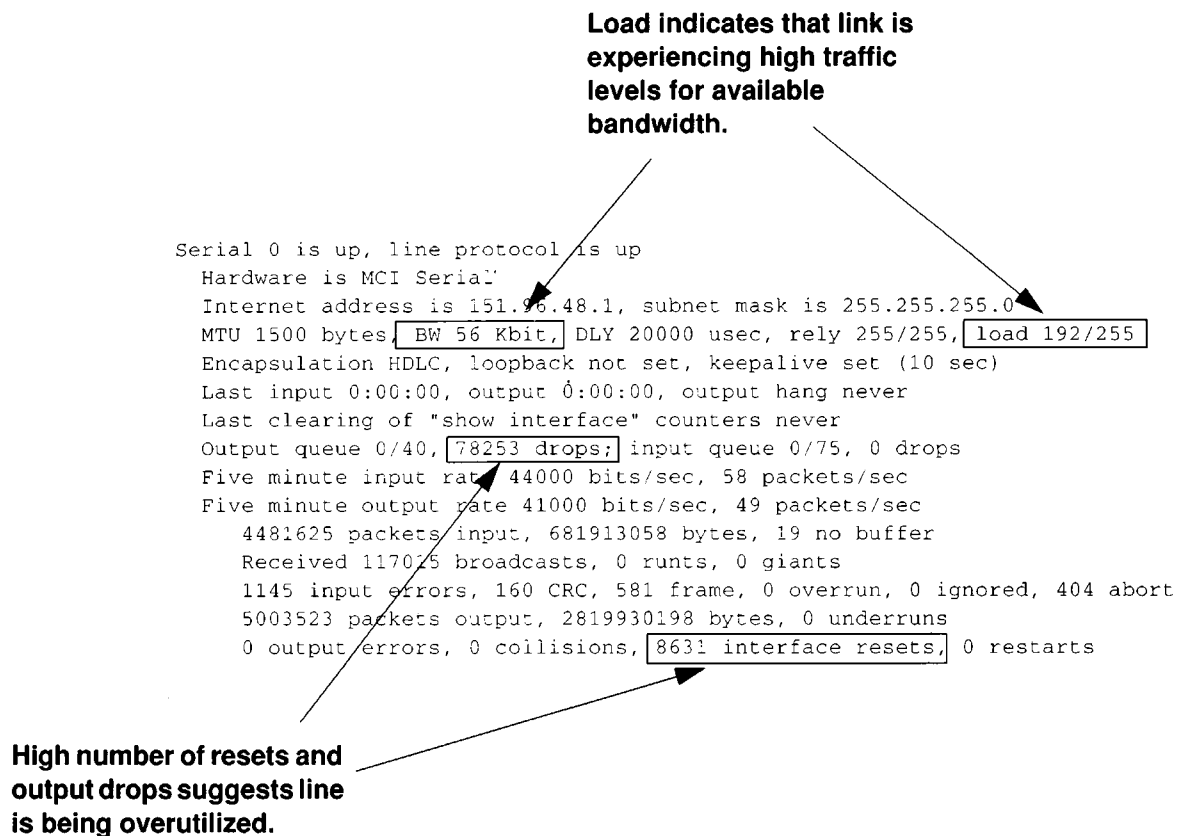


Figure 8-13 Display Output of Show Interfaces Command

Of interest in this display is the fact that the value for input errors is relatively low, but the values for interface resets and output drops are both high. Another clue is that the load field indicates that the link is experiencing a load of about 75 percent of available bandwidth. This information combines to suggest that the serial line is functional, but is being overutilized.

Note: Use care when reading the value specified in the “load” field displayed by the **show interfaces serial** command. This is only a gross measure of traffic activity on the interface. In addition, the value displayed (and calculated) is based on the number shown in the “BW” field on the same line. The BW value displayed and used in the load calculation normally adopts a default value (typically 56 Kbps or 1544 Kbps) that depends on the type of interface installed. If the actual available bandwidth differs from this default value, you must explicitly enter the actual bandwidth using the **bandwidth** interface subcommand for load to be calculated correctly.

To monitor changes in the number of dropped packets, follow this brief sequence:

- Obtain **show interfaces serial** display (as shown in Figure 8-13)
- Write down the number of output drops (78253 in Figure 8-13)

- Use the **clear counters** command to reset counters on the target interface.
- Check the change to the output drops field in an hour; if the value is around 1000 or more, the link is probably overutilized.

Note: DEC's DECnet protocol is particularly sensitive to drops. If your internet involves handling of DECnet traffic, you must ensure that drops are eliminated.

To further confirm that the serial link is being overutilized, use the **show buffers** command. Figure 8-14 illustrates the output from this command. In this example, there are a large number of failures and misses. This suggests that there is some kind of problem with the system-level buffers and that the traffic the routers are trying to transmit exceeds the interface bandwidth.

Note: An interface reset on one end of a serial link causes aborts on the other end; these appear as input errors (as well as aborts). This is why it is essential to inspect both ends of the serial link (using the **show interfaces** command).

```

Buffer elements:
  500 in free list (500 max allowed)
  19384 hits, 0 misses, 0 created
Small buffers, 104 bytes (total 120, permanent 120):
  120 in free list (0 min, 250 max allowed)
  986320 hits, 0 misses, 0 trims, 0 created
Middle buffers, 600 bytes (total 93, permanent 90):
  90 in free list (0 min, 200 max allowed)
  3187056 hits, 17831 misses, 11049 trims, 11052 created
Big buffers, 1524 bytes (total 90, permanent 90):
  90 in free list (0 min, 120 max allowed)
  345109 hits, 0 misses, 0 trims, 0 created
Large buffers, 5024 bytes (total 5, permanent 5):
  5 in free list (0 min, 30 max allowed)
  0 hits, 0 misses, 0 trims, 0 created
Huge buffers, 18024 bytes (total 0, permanent 0):
  0 in free list (0 min, 4 max allowed)
  0 hits, 0 misses, 0 trims, 0 created

17831 failures (0 no memory)

```

High miss and failure counts confirm suspicion that link is overutilized.

Figure 8-14 Show Buffers Command Output

Step 2: The router configuration files are the next place to look for clues in this scenario. If fast switching is *not* explicitly disabled for all protocols, this is the first configuration change to try. Fast switching is enabled by default. For DECnet, disable fast switching with the **no decnet route-cache** interface subcommand. This change forces the router to use system-level memory buffers (instead of board-level buffers), which under these conditions, can improve overall throughput.

Step 3: Although disabling fast switching can improve performance over the serial link, assume that problems still persist during peak demand times. The next step is to prioritize traffic using the priority queuing function. By assigning a “high” priority to bridged (LAT) packets, the LAT traffic takes precedence over any other traffic. Again, this enhances performance, but might not entirely eliminate peak period sluggishness.

Step 4: The final action in this case is to upgrade software to a more recent software version (for example, from revision 8.2 to 9.0). Software Releases 8.3 and higher allow administrators to configure specific buffer sizes. By setting a minimum number of system buffers as available at all times, it is possible to significantly reduce the bottleneck at the serial link.

Figure 8-15 illustrates a complete configuration listing for Router-Home (obtained using the **write terminal** command) that includes the changes suggested in Steps 2 through 4.

Note: When customizing buffer settings, you must carefully match the buffer specifications with the hold-queue limits. Refer to Chapter 7, “Troubleshooting WAN Connectivity,” for more information about managing buffers and specifying hold-queue limits.

```

Current configuration:
!
enable-password ZpYptZ
!
!
buffers small min-free 20
buffers middle min-free 20
buffers big min-free 5
buffers small max-free 300
buffers middle max-free 400
!
!
decnet routing 21.12
decnet node area
decnet max-address 1023
!
!
interface Ethernet 0
ip address 129.14.87.123 255.255.255.0
decnet cost 5
bridge-group 1
!
interface Serial 0
ip address 151.96.48.1 255.255.255.0
no ip route-cache
decnet cost 20
no decnet route-cache
bridge-group 1
priority-group 1
!
!
router igrp 109
network 129.14.0.0
network 151.96.0.0
!
!
!
ip name-server 255.255.255.255
snmp-server community
snmp-server community public RO
hostname Router-Home
scheduler-interval 1500
bridge 1 protocol dec
!
priority-list 1 protocol bridge high list 201
priority-list 1 protocol decnet medium
priority-list 1 protocol ip normal
priority-list 1 queue-limit 40 40 20 10
!
!
access-list 201 permit 0x6004 0x0000
!
!
end

```

Figure 8-15 Complete Configuration Showing Changes Needed

Problem Solution Summary

Clearly, this case revolved around an interface that was overworked. The immediate reaction to this situation might be to add another link in parallel. Ultimately, adding bandwidth is probably required. But that might not be an immediately available option. Perhaps the protocol being used cannot handle load balancing, or you simply cannot afford the added expense of another physical link within your current budget.

The actions offered in this example explore options that use the existing physical configuration, but reconfigure the way traffic is handled. To recap, the following modifications can help optimize traffic over an overloaded 56-Kbps link:

- Disable fast switching.
- Enable priority queuing for bridged (LAT) traffic. Note that an access list is included with this configuration that permits the bridging of LAT packets, but blocks bridging of any other packets.

The configuration also shows specific queue depths for high, medium, normal, and low priority packets. If you implement priority queuing, try these as a starting point, but your actual implementation will take some tuning. With LAT, reducing the number of drops will improve performance. However, avoid assigning arbitrarily large queue limits, because there can be performance side effects. Queues that are excessively large can cause timing problems as a result of specific packets being buffered too long.

As you tune the **queue-limit** values assigned in the **priority-list** global command, check the interface activity with the **show interfaces** command to monitor the number of drops. A typical **queue-limit** value for the high-priority packets is 50.

This solution is particularly applicable to situations involving routing of DECnet traffic and bridging of LAT traffic. Set DECnet queues to 100 to help prevent drops if necessary.

- Upgraded router software to Software Release 8.3 or higher and customize use of system buffers.

Note: As a final note, if an interface is dropping packets at a traffic load of 100 percent (255/255 load), you need to add bandwidth in the form of another or a higher-capacity line.

Chapter 9

Troubleshooting Internet Performance

9

Poor Internetwork Performance Symptoms 9-2

Sporadic Service Availability and Poor AppleTalk Internet Performance 9-3

Possible Causes and Suggested Actions 9-3

Slow Performance and Intermittent Loss of Connections over RSRB 9-5

Possible Causes and Suggested Actions 9-5

Poor Novell Server Performance over Router in LAN Internet 9-6

Possible Causes and Suggested Actions 9-6

Poor Novell Server Performance over Router in WAN 9-7

Possible Causes and Suggested Actions 9-7

Generally Slow Performance in TCP/IP Internetworks 9-8

Possible Causes and Suggested Actions 9-8

Slow TCP/IP Performance Despite Multiple Paths 9-9

Possible Causes and Suggested Actions 9-9

Load Balancing Problem Example 9-10

Slow Host or Network Response over WAN or Serial Link 9-11

General Diagnostic Information 9-11

Possible Causes and Suggested Actions 9-11

Loss of Connections over WAN or Serial Link 9-13

Possible Causes and Suggested Actions 9-13

Chapter 9

Troubleshooting Internet Performance

9

This chapter focuses on common symptoms associated with poor performance in internetworks, possible causes of those symptoms, and general suggestions for identifying, isolating, and resolving causes.

This chapter consists of the following sections:

- Performance symptom list
- Symptom/cause/action modules

The symptom/cause/action modules consist of the following sections:

- Symptom statement—A specific symptom associated with the technology/media/protocol in which this module appears.
- Possible causes and suggested actions—A table for each symptom containing possible causes for the symptom and suggested actions for resolving each cause.



Caution: Throughout this and other chapters, the use of **debug** commands is suggested for obtaining information about network traffic and router status. Use these commands with great care. In general, it is recommended that these commands only be used under the direction of your router technical support representative when troubleshooting specific problems. Enabling debugging can disrupt operation of the router when internets are experiencing high load conditions. When you finish using a **debug** command, remember to disable it with its specific **undebug** command or with the **undebug all** command.

Poor Internetwork Performance Symptoms

The symptom modules that follow pertain to performance-related problems for the various protocols and technologies addressed in this publication. Unless otherwise indicated, each module is presented as a set of problems associated with a specific technology or protocol. Where there are special considerations associated with a specific network type, notes are included.

Performance symptoms discussed in this section include the following:

- Sporadic service availability and poor AppleTalk internet performance
- Slow performance and intermittent loss of connections over Cisco RSRB
- Poor Novell server performance over router in LAN internet
- Poor Novell server performance over router in WAN
- Generally slow performance in TCP/IP internetworks
- Slow TCP/IP performance despite multiple paths
- Slow network or host response over WAN or serial link
- Loss of connections over WAN or serial link

Sporadic Service Availability and Poor AppleTalk Internet Performance

Symptom: Connectivity to AppleTalk services over an internetwork is unpredictable and generally slow.

Possible Causes and Suggested Actions

Table 9-1 outlines possible causes of unpredictable and slow performance in an AppleTalk internetwork.

Table 9-1 Causes and Actions for Poor AppleTalk Internet Performance

Possible Cause	Suggested Actions
ZIP storm	<p>Step 1: Use show appletalk traffic to look for number of ZIP Requests displayed; repeat after 30 seconds or so.</p> <p>Step 2: Compare resulting display output. If number is greater than 10 and increasing, there is probably a ZIP storm occurring.</p> <p>Step 3: Use show appletalk route to see whether a network shows up in the table, even though the zone indicates no zone set in the display.</p> <p>Step 4: If you find a network with this zone specification, a node in that network is probably not responding to ZIP requests, resulting in the ZIP storm.</p> <p>Step 5: Determine why the node is not responding to ZIP requests.</p>
Duplicate network numbers	<p>Step 1: The network where these symptoms are noticeable is likely to contain duplicate network numbers equidistant from the point where problems are observed.</p> <p>Either change the network number of the afflicted network or remove Appletalk from the suspect/problem interface. In either case, the interface's original network number should disappear from the internet within a few minutes. If it persists, you probably have found the duplicate network.</p> <p>Step 2: If you changed the network number on the interface, no further action is required. If not, change it now (make sure it is unique). Remember to reenter the zone name and any other interface configurations for Appletalk on that interface.</p>

Possible Cause	Suggested Actions
Unexpected back door	<p><i>Step 1:</i> Inspect internetwork for any bridges that may link networks that are routing AppleTalk.</p> <p><i>Step 2:</i> If any bridges (including routers configured for bridging) are found, set all bridges to forward nonrouted protocols and filter routed protocols.</p> <p><i>Step 3:</i> Monitor reporting of routes and neighbors with show appletalk route and show interfaces commands.</p> <p><i>Step 4:</i> If networks continue to be associated with the wrong interfaces, consult your router technical support representative for more assistance.</p>

Slow Performance and Intermittent Loss of Connections over RSRB

Symptom: Users complain about connection loss at peak traffic periods when trying to connect to resources on the other side of a router configured for remote source-route bridging (RSRB).

Possible Causes and Suggested Actions

Table 9-2 outlines possible causes and suggested actions when users experience intermittent connectivity in RSRB interconnections.

Table 9-2 Causes and Actions for Load-Related RSRB Performance Problems

Possible Cause	Suggested Actions
Busy router; high CPU utilization when using TCP encapsulation	Step 1: Use show process EXEC command to determine CPU utilization. Look for CPU utilization higher than 50 percent. High CPU utilization can cause RSRB sessions to time out when using TCP encapsulation.
	Step 2: Check configuration for keyword local-ack at the end of the source-bridge remote-peer global configuration command.
	Step 3: Add this additional keyword if missing.
	Step 4: Consider implementing Fast Sequenced Transport (FST) on the link, using the source-bridge fst-peername global configuration command. Refer to your <i>Router Products Configuration and Reference</i> publication for details about FST.

Poor Novell Server Performance over Router in LAN Internet

Symptom: Users complain about sessions dropping at peak traffic periods when trying to connect to resources on the other side of a router configured to route Novell IPX.

Possible Causes and Suggested Actions

Table 9-3 outlines possible causes of poor file server response in a routed Novell IPX LAN internetwork.

Table 9-3 Causes and Actions for Novell Performance Problems in LAN Internet

Possible Cause	Suggested Actions
Excessive traffic; collisions causing session drops (Ethernet problem only)	Step 1: Use protocol analyzer to examine traffic.
	Step 2: Look for collisions in excess of normal acceptable conditions (varies for specific site). As an alternative, use the show interfaces EXEC command for a rough estimate of collision count.
	Step 3: Examine output of protocol analyzer to determine bandwidth utilization. Analyzer information will provide more accurate reading of dynamic traffic information. As an alternative, you can use a Novell server's load monitor command at the server console prompt to get an approximate idea of bandwidth utilization.
	Step 4: If bandwidth utilization detected by the analyzer is 15 to 20 percent (on average) or higher, you are likely to have a load-related performance problem.
	Step 5: If you see that collisions are increasing steadily with a higher-than-expected bandwidth utilization, consider segmenting the network with additional bridges or routers.
Insufficient bandwidth on Token Ring to handle traffic	Step 1: Upgrade from 4-Mbps to 16-Mbps Token Ring throughout network.
	Step 2: If performance is still inadequate, consider segmenting the network with additional bridges or routers.

Poor Novell Server Performance over Router in WAN

Symptom: Users complain about sessions dropping at peak traffic periods when trying to connect to resources on the other side of a router configured to route Novell IPX over WAN or serial link.

Possible Causes and Suggested Actions

Table 9-4 outlines possible causes of poor file server response in a routed Novell IPX internetwork.

Table 9-4 Causes and Actions for Novell Performance Problems in WAN

Possible Cause	Suggested Actions
Other protocol dominates CPU time	<p>Step 1: Use the show process command to look for large numbers appearing in the “Runtime (ms)” and “Invoked” fields for certain protocols. An example would be a protocol that has a value that is 10 times or greater than the value indicated for Novell traffic.</p> <p>When this kind of condition exists, Novell traffic is not getting adequate access to the CPU, and performance is affected.</p> <p>Step 2: Use the show interfaces command to look for a high level of output drops.</p> <p>Step 3: If you see output drops and a particular other protocol is dominating CPU time (per show process “Runtime (ms)” field), use priority queuing to force system to handle Novell traffic over other protocols. More information about priority queuing is provided in Chapter 7, “Troubleshooting WAN Connectivity.”</p> <p>Step 4: If priority queuing does not work, add bandwidth by implementing a higher-speed line or adding additional lines (of same speed).</p>

Generally Slow Performance in TCP/IP Internetworks

Symptom: TCP/IP internetwork performance is slow, with poor host response, spotty connection service, and generally slow file transfers. Packets may be dropped.

Possible Causes and Suggested Actions

Table 9-5 outlines possible causes of generally slow performance in a TCP/IP internetwork.

Table 9-5 Causes and Actions for Slow Performance in TCP/IP Internets.

Possible Cause	Suggested Actions
Bad network link; results in packets being dropped and lost	Step 1: Ping out along entire length of path to determine where packets are being dropped.
	Step 2: Refer to Chapter 1 for general hardware diagnostic information. Refer to Chapter 7, "Troubleshooting WAN Connectivity," for more specific information about serial debugging.
	Step 3: Perform serial debugging or other media debugging.
	Step 4: Replace hardware or add bandwidth as necessary.
Access list applied to one link but not another (when there are multiple paths to a destination)	Step 1: Refer to the symptom section following this section entitled "Slow TCP/IP Performance Despite Multiple Paths."
Congested link	Step 1: Determine whether the link is indeed congested.
	Step 2: Refer to Chapter 1 for general diagnostic information. Refer to Chapter 7, "Troubleshooting WAN Connectivity," for more information about serial debugging.
	Step 3: Apply priority queuing if feasible.
	Step 4: Add bandwidth or additional routers if priority queuing does not help.

Slow TCP/IP Performance Despite Multiple Paths

Symptom: Despite multiple paths from one network to another and apparently sufficient bandwidth, performance over the links is poor and traffic does not appear to be getting through some of the links. Although this can be considered a connectivity problem, it manifests itself as a performance issue.

Possible Causes and Suggested Actions

Table 9-6 outlines possible causes of performance problems in TCP/IP networks because some paths are being blocked.

Table 9-6 Causes and Actions for Poor Performance Because of Blocked Paths

Possible Cause	Suggested Actions
Misconfigured access lists where there are multiple paths and one or more access lists block access to one or more routes	Step 1: Use the ping and trace EXEC commands to determine where traffic is stopping. Works best when standard access lists are used.
	Step 2: If ping or trace packets are stopped along the way, check the specific router for access lists.
	Step 3: If an access list is found, disable the list and monitor traffic through the router, using the ping and trace commands.
	Step 4: If ping and trace packets get through after removing the access list, you might need to add explicit permit statements to the access list to allow blocked traffic type. If extended access lists are specified, ping and trace packets might get through even though intended traffic is not getting through.
	Step 5: If ping packets get through, use a <i>Sniffer</i> along the path where problems occur to see where the dropped packet type was last seen. The next node is the most likely suspect.
	Step 6: As in the prior access list discussion, remove the access list and monitor traffic through the router using the protocol being blocked.
Bad interface or media hardware	Step 1: Refer to Chapter 1 for general hardware diagnostic information. Refer to Chapter 7, "Troubleshooting WAN Connectivity," for more information about serial debugging.
	Step 2: Perform serial debugging or other media debugging.
	Step 3: Replace hardware or add bandwidth as necessary.
Load balancing problem (see Figure 9-1 that follows this table)	Step 1: Use show interfaces and show ip traffic EXEC commands and ping out to destination to determine where traffic is being dropped.
	Step 2: At point of congestion, relieve traffic problems by adding a router in parallel or by increasing the bandwidth of the link.

Possible Cause**Suggested Actions**

Step 3: If you cannot add a router or bandwidth, try the following:

On the congested router, adjust the hop count by using the **offset-list** command to add a hop to a route received from a particular router.

Use the **distance** router subcommand to set the administrative distance for a particularly slow route.

Load Balancing Problem Example

Figure 9-1 illustrates a situation where two routes may be equivalent in terms of hop count from Host-D to Far-Net, but due to the level of traffic (associated with Router-B and the Data Center), the alternative route (through Router-A) is administratively preferred. In this case, both routes look equally good to Host-D, so without any configuration modifications, Host-D can use either Router-A or Router-B to communicate with Far-Net. However, as outlined in the load balancing problem discussion in Table 9-6, several options are available to force traffic from Host-D (intended for Far-Net) to go through Router-A.

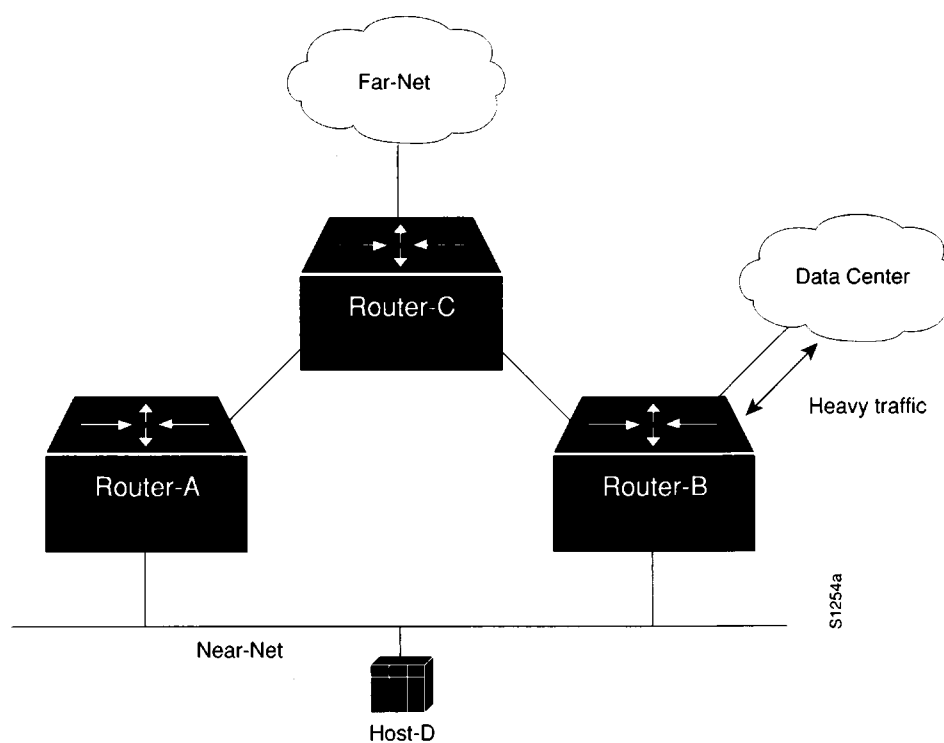


Figure 9-1 Load Balancing Problem Map

Slow Host or Network Response over WAN or Serial Link

Symptom: As with similar loss of connection problems, users complain about very slow host and network responsiveness at peak traffic periods over a WAN or serial link.

General Diagnostic Information

In general, obtaining the following information will be useful in troubleshooting load-related connection problems:

- Observe output of **show interface serial** command on both ends of the serial line; evaluate error counters.
- If you do see input errors, refer to “Evaluating Input Errors” in Chapter 7, “Troubleshooting WAN Connectivity,” for details about isolating the sources of input errors.

Possible Causes and Suggested Actions

Table 9-7 outlines possible causes of load-related performance in serial and WAN interconnections.

Table 9-7 Causes and Actions for Load-Related WAN Performance Problems

Possible Cause	Suggested Actions
Dirty serial line	<i>Step 1:</i> Determine whether input errors are increasing. <i>Step 2:</i> If input errors appear, diagnose serial line per discussion in Chapter 7, “Troubleshooting WAN Connectivity.”
Overutilized bandwidth	<i>Step 1:</i> If input errors do not appear, the problem is related to congestion. <i>Step 2:</i> Turn off fast switching on affected interface. <i>Step 3:</i> Check applications being run, especially for very large file transfers scheduled at particular times of day. <i>Step 4:</i> If this is the case, set up priority queue (requires that the protocol allows flow control). <i>Step 5:</i> Rearrange file transfer timing by applications so that links are not overused during normal business hours. <i>Step 6:</i> Add bandwidth and consider using dial backup over the new link for applications that are taking excessive bandwidth on existing links. <i>Step 7:</i> Adjust buffer size (8.3 or more recent software needed).

Possible Cause	Suggested Actions	
Hardware in the serial link is unreliable	<i>Step 1:</i>	Troubleshoot serial line per CSU/DSU loopback tests and ping tests described in Chapter 7 or with a serial analyzer.
	<i>Step 2:</i>	Replace hardware as necessary.
Carrier is automatically rerouting T1 trunk lines	<i>Step 1:</i>	Contact long line carrier service to determine whether this is happening.
	<i>Step 2:</i>	Ensure that carrier provides dedicated circuit if automatic switching is causing performance problems.

Loss of Connections over WAN or Serial Link

Symptom: Users complain about dropped connections and the inability to make host connections at peak traffic periods. One example of this problem is in an environment featuring bridged DEC Local Area Transport (LAT) traffic and multiple routed protocols. Data entry input from users (or other application requests) may be getting buffered at the end of an already long input queue—eventually one end of the connection will time out.

Possible Causes and Suggested Actions

Table 9-8 outlines possible causes of load-related connection drops in serial and WAN interconnections.

Table 9-8 Causes and Actions for WAN Performance-Related Loss of Connections

Possible Cause	Suggested Actions
Dirty serial line	<p><i>Step 1:</i> Determine whether input errors are increasing.</p> <p><i>Step 2:</i> If input errors appear, diagnose serial line per discussion in Chapter 7, “Troubleshooting WAN Connectivity.”</p>
Overutilized bandwidth	<p><i>Step 1:</i> If input errors do not appear, the problem is related to congestion.</p> <p><i>Step 2:</i> Turn off fast switching on affected interface.</p> <p><i>Step 3:</i> Check applications being run, especially for very large file transfers scheduled at particular times of day.</p> <p><i>Step 4:</i> If this is the case, set up priority queue (requires that the protocol allows flow control).</p> <p><i>Step 5:</i> When bridging LAT, consider implementing LAT compression to reduce bandwidth.</p> <p>Use the bridge-group group lat-compression interface subcommand.</p> <p><i>Step 6:</i> Rearrange file transfer timing by applications so that links are not overused during normal business hours.</p> <p><i>Step 7:</i> Add bandwidth and consider using dial backup over the new link for applications that are taking excessive bandwidth on existing links.</p> <p><i>Step 8:</i> Adjust buffer size (8.3 or more recent software needed).</p>
Hardware in the serial link is unreliable	<p><i>Step 1:</i> Troubleshoot serial line per CSU/DSU loopback tests and ping tests described in Chapter 7 or with a serial analyzer.</p>
Inadequate bandwidth	<p><i>Step 1:</i> After checking all of the above, inspect the show interfaces serial display.</p> <p><i>Step 2:</i> If after these actions, load is still indicated at about 80 percent, your line is inadequate for traffic requirements.</p> <p><i>Step 3:</i> Add another serial line.</p>



Chapter 10

Debug Command Reference

10

General Debugging Information 10-1

- Using Debug Commands 10-1
- Using the Debug ? Command 10-2
- Using the Debug All Command 10-2
- Generating Debugging Command Output 10-2
- Redirecting Debugging and Error Message Output 10-3
 - Enabling Message Logging 10-4
 - Logging Messages to the Console 10-4
 - Logging Messages to an Internal Buffer 10-5
 - Logging Messages to Another Monitor 10-5
 - Logging Messages to a UNIX Syslog Server 10-6
 - Limiting Messages to a Syslog Server 10-6

Debug Command Listing 10-8

- Debug Apple-ARP 10-9
- Debug Apple-Errors 10-10
- Debug Apple-Events 10-12
- Debug Apple-NBP 10-16
- Debug Apple-Packet 10-19
- Debug Apple-Routing 10-21
- Debug Apple-ZIP 10-23
- Debug ARP 10-24
- Debug Broadcast 10-25
- Debug DECnet-Connects 10-27
- Debug Frame-Relay 10-28
- Debug Frame-Relay-Events 10-30
- Debug Frame-Relay-LMI 10-31
- Debug Frame-Relay-Packets 10-34
- Debug IP-ICMP 10-36
- Debug IP-IGRP 10-40
- Debug IP-IGRP-Events 10-42
- Debug IP-OSPF-Events 10-43
- Debug IP-Packet 10-44
- Debug IP-RIP 10-47
- Debug IP-TCP 10-48
- Debug LAPB 10-50

- Debug LNM-Events 10-54
- Debug LNM-LLC 10-56
- Debug LNM-MAC 10-59
- Debug Local-ACK-State 10-61
- Debug Novell-Packet 10-62
- Debug Novell-Routing 10-63
- Debug Novell-SAP 10-64
- Debug Packet 10-68
- Debug RIF 10-70
- Debug Serial-Interface 10-74
 - Debug Serial-Interface for DDR 10-75
 - Debug Serial-Interface for Frame Relay Encapsulation 10-75
 - Debug Serial-Interface for HDLC 10-76
 - Debug Serial-Interface for HSSI 10-78
 - Debug Serial-Interface for ISDN Basic Rate 10-79
 - Debug Serial-Interface for an MK5025 Device 10-80
 - Debug Serial-Interface for PPP Encapsulation 10-81
 - Debug Serial-Interface for SMDS Encapsulation 10-82
- Debug Serial-Packet 10-83
 - Debug Serial-Packet for DDR 10-83
 - Debug Serial-Packet for PPP 10-83
 - Debug Serial-Packet for SMDS Encapsulation 10-85
- Debug Source-Event 10-86
- Debug Span 10-91
 - IEEE Spanning Tree Example 10-91
 - DEC Spanning Tree Example 10-92
- Debug TFTP 10-94
- Debug Token-Ring 10-95
- Debug VINES-ARP 10-98
- Debug VINES-Echo 10-99
- Debug VINES-Packet 10-100
- Debug VINES-Routing 10-101
- Debug VINES-Table 10-102
- Debug XNS-Packet 10-103
- Debug XNS-Routing 10-104
- Debug X25 10-105
- Debug X25-Events 10-109
- Debug X25-VC 10-110

Chapter 10

Debug Command Reference

10

This chapter covers the **debug** commands that you can use to diagnose and resolve internetworking problems. This chapter includes two sections:

- General debugging information
- Debug command listing



Caution: Because debugging output is assigned high priority, excessive debugging output can render the system unusable. For this reason, you should only use **debug** commands to troubleshoot specific problems or during troubleshooting sessions with Cisco technical support staff. Moreover, it is best to use **debug** commands during off hours, when lower network traffic and fewer users reduce the risk of using them.

General Debugging Information

This section covers the following topics:

- Using **debug** commands
- Using the **debug ?** command
- Using the **debug all** command
- Generating debugging output
- Redirecting debugging output

Using Debug Commands

You can use the privileged EXEC command **debug** to display several classes of network events on the console terminal.

Most **debug** commands do not take any required or optional arguments. For example, to enable the **debug broadcast** command, enter the following in privileged mode at the command line:

```
debug broadcast
```

To turn off the **debug broadcast** command, enter the following in privileged mode at the command line:

undebug broadcast

To display the state of each debugging option, enter the following at the command line:

show debugging

Note: Throughout this chapter, it is assumed that a particular **debug** command takes no arguments unless otherwise noted.

Using the Debug ? Command

To list and briefly describe all of the debugging command options, enter the **debug ?** command in privileged mode at the command line.

debug ?

Using the Debug All Command

To enable all system diagnostics, enter the following in privileged mode at the command line:

debug all

Its converse, the **undebug all** command, turns off all diagnostic output. The **undebug all** command is a convenient way to ensure that you have not accidentally left any **debug** commands turned on.



Caution: Because debugging output takes priority over other network traffic, and because the **debug all** command generates more output than any other **debug** command, it can severely diminish the router's performance or even render it unusable. In virtually all cases, it is best to use more specific **debug** commands.

Generating Debugging Command Output

Turning on a debugging command can result in output similar to the example for the **debug broadcast** command shown in Figure 10-1.

```
Ethernet0: Broadcast ARPA, src 0000.0c00.6fa4, dst ffff.ffff.ffff, type 0x0800,
data 4500002800000000FF11EA7B, len 60
Serial3: Broadcast HDLC, size 64, type 0x800, flags 0x8F00
Serial2: Broadcast PPP, size 128
Serial7: Broadcast FRAME-RELAY, size 174, type 0x800, DLCI 7a
Ultranet0: Broadcast ULTRANET, size 60
```

Figure 10-1 Example Debug Broadcast Output

The router continues to generate such output until you enter the converse **undebug** command (in this case, **undebug broadcast**).

If you enable a particular **debug** command and no output is displayed, consider the following possibilities:

- The router may not be properly configured to generate the type of traffic you hope to monitor. Use the **write terminal** command to check its configuration.
- Even if the router is properly configured, it may not generate any of the type of traffic you hope to monitor during the particular period that debugging is turned on. Depending on the protocol you are debugging, you can use commands such as the TCP/IP command **ping** to generate network traffic.

Redirecting Debugging and Error Message Output

By default, the network server sends the output from **debug** commands and system error messages to the console terminal. If you accept this default, it is best to monitor debugging output using a VTY connection, rather than the console port.

To redirect debugging output, use the **logging** command options within configuration mode.

Possible destinations include the console terminal, virtual terminals, and UNIX hosts running a syslog server; the syslog format is compatible with 4.3 BSD UNIX and its derivatives.

Note: Be aware that the debugging destination you use impacts system overhead. Logging to the console produces very high overhead, whereas logging to VTY produces less overhead. Logging to a syslog server produces even less, whereas logging to memory produces the least overhead of any method.

To configure message logging, you need to be in the configuration command collection mode. To enter this mode, use the EXEC command **configure terminal** at the EXEC prompt. The following sections describe how to implement these redirection options.

Enabling Message Logging

To enable message logging to all supported destinations other than the console, enter the following:

logging on

This behavior is the default.

To enable logging to the console terminal only, enter the following:

no logging on

Logging Messages to the Console

To limit the types of messages that are logged to the console, use the **logging console** global configuration command. The full syntax of this command follows.

logging console *level*

no logging console

The **logging console** command limits the logging messages displayed on the console terminal to messages with a level number at or below the specified severity level, which is specified by the *level* argument.

The argument *level* can be one of the keywords listed in Table 10-1. They are listed in order from the most severe to the least severe level.

Table 10-1 Logging Message Keywords and Levels

Level	Keyword	Description	Syslog Definition
0	emergencies	System is unusable.	LOG_EMERG
1	alerts	Immediate action is needed.	LOG_ALERT
2	critical	Critical conditions exist.	LOG_CRIT
3	errors	Error conditions exist.	LOG_ERR
4	warnings	Warning conditions exist.	LOG_WARNING
5	notification	Normal, but significant, conditions exist.	LOG_NOTICE
6	informational	Informational messages.	LOG_INFO
7	debugging	Debugging messages.	LOG_DEBUG

The **no logging console** command disables logging to the console terminal.

Example

This command sets console logging of messages at the debugging level:

```
!  
logging console debugging  
!
```

Logging Messages to an Internal Buffer

The default logging device is the console; all messages are displayed on the console unless otherwise specified.

To log messages to an internal buffer, use the **logging buffered** global configuration command. The full command syntax follows.

logging buffered
no logging buffered

The **logging buffered** command copies logging messages to an internal buffer instead of writing them to the console terminal. The buffer is circular in nature, so newer messages overwrite older messages. To display the messages that are logged in the buffer, use the EXEC command **show logging**. The first message displayed is the oldest message in the buffer.

The **no logging buffered** command cancels the use of the buffer and writes messages to the console terminal (the default).

Logging Messages to Another Monitor

To limit the level of messages logged to the terminal lines (monitors), use the **logging monitor** global configuration command. The full syntax of this command follows.

logging monitor level
no logging monitor

The **logging monitor** command limits the logging messages displayed on terminal lines other than the console line to messages with a level at or above the value of the *level* variable. The argument *level* is one of the keywords described for the **logging console** command in a previous section, "Logging Messages to the Console." To display logging messages on a terminal, use the privileged EXEC command **terminal monitor**.

The **no logging monitor** command disables logging to terminal lines other than the console line.

This command sets the level of messages displayed on monitors other than the console to notification:

```
!  
logging monitor notification  
!
```

Logging Messages to a UNIX Syslog Server

To log messages to the syslog server host, use the **logging** global configuration command. The full syntax is as follows:

```
logging internet-address  
no logging internet-address
```

The **logging** command identifies a syslog server host to receive logging messages. The argument *internet-address* is the Internet address of the host. By issuing this command more than once, you build a list of syslog servers that receive logging messages.

The **no logging** command deletes the syslog server with the specified address from the list of syslogs.

Limiting Messages to a Syslog Server

To limit how many messages are sent to the syslog servers, use the **logging trap** global configuration command. Its full syntax follows.

```
logging trap level  
no logging trap
```

The **logging trap** command limits the logging messages sent to syslog servers to messages with a level at or above the value of the *level* variable. The argument *level* is one of the keywords described for the **logging console** command in Table 10-1.

To send logging messages to a syslog server, specify its host address with the **logging** command.

The default trap level is **informational**.

The **no logging trap** command disables logging to syslog servers.

The current software generates four categories of syslog messages:

- Error messages about software or hardware malfunctions, displayed at the **errors** level.
- Interface up/down transitions and system restart messages, displayed at the **notification** level.
- Reload requests and low-process stack messages, displayed at the **informational** level.
- Output from the **debug** commands, displayed at the **debugging** level.

The EXEC command **show logging** displays the addresses and levels associated with the current logging setup. The command output also includes ancillary statistics.

Example

To set up the syslog daemon on a 4.3 BSD UNIX system, include a line such as the following in the file */etc/syslog.conf*:

```
local7.debug /usr/adm/logs/tiplog
```

The **local7** keyword specifies the logging facility to be used.

The *debug* argument specifies the syslog level. See the previous *level* arguments list for other arguments that can be listed.

The UNIX system sends messages at or below this level to the file specified in the next field. The file must already exist, and the syslog daemon must have permission to write to it.

Debug Command Listing

This section includes an alphabetical listing of **debug** commands. Documentation for each command includes a description of its use, example output, and a description of that output.

Output formats of the various **debug** commands vary. Some generate a single line of output per packet, whereas others generate multiple lines of output per packet. Some generate a great deal of output, whereas others generate occasional output. Some generate lines of text; others generate information in field format. Thus, the way the **debug** commands are documented also varies. For example, for **debug** commands that generate lines of text, the output is described line-by-line. For **debug** commands that generate output in field format, tables are used to describe the fields.

Debug Apple-ARP

Use the **debug apple-arp** command to enable debugging of the AppleTalk address resolution protocol.

This command is helpful when you experience problems communicating with a node on the network you control (neighbor). If the **debug apple-arp** display indicates that the router is receiving AARP probes, you can assume that the problem does not reside at the physical layer.

Note: A side effect of enabling this command is that gleaned MAC information from datagrams is disabled.

Figure 10-2 shows example **debug apple-arp** output.

```
Ether0: AARP: Sent resolve for 4160.26
Ether0: AARP: Reply from 4160.26(0000.0c00.0453) for 4160.154(0000.0c00.8ea9)
Ether0: AARP: Resolved waiting request for 4160.26(0000.0c00.0453)
Ether0: AARP: Reply from 4160.19(0000.0c00.0082) for 4160.154(0000.0c00.8ea9)
Ether0: AARP: Resolved waiting request for 4160.19(0000.0c00.0082)
Ether0: AARP: Reply from 4160.19(0000.0c00.0082) for 4160.154(0000.0c00.8ea9)
```

Figure 10-2 Example Debug Apple-ARP Output

Explanations for representative lines of output in Figure 10-2 follow.

The following line of output indicates that the router has requested the hardware MAC address of the host at network address 4160.26.

```
Ether0: AARP: Sent resolve for 4160.26
```

The following line of output indicates that the host at network address 4160.26 has replied, giving its MAC address (0000.0c00.0453). For completeness, the message also shows the network address to which the reply was sent and its hardware MAC address (also in parentheses).

```
Ether0: AARP: Reply from 4160.26(0000.0c00.0453) for 4160.154(0000.0c00.8ea9)
```

The following line of output indicates that the MAC address request is complete.

```
Ether0: AARP: Resolved waiting request for 4160.26(0000.0c00.0453)
```

Debug Apple-Errors

Use the **debug apple-errors** command to display errors occurring in the AppleTalk network.

To solve encapsulation problems, enable **debug apple-errors** and **debug apple-packet** together.

Note: In a stable AppleTalk network, **debug apple-errors** should produce little output.

Figure 10-3 shows example **debug apple-errors** output when a router is brought up with a zone that does not agree with the zone list of other routers on the network.

```
%AT-3-ZONEDISAGREES: Ethernet0: AppleTalk port disabled; zone list incompatible with 4160.19
%AT-3-ZONEDISAGREES: Ethernet0: AppleTalk port disabled; zone list incompatible with 4160.19
%AT-3-ZONEDISAGREES: Ethernet0: AppleTalk port disabled; zone list incompatible with 4160.19
```

Figure 10-3 Example Debug Apple-Errors Output

As Figure 10-3 suggests, a single error message indicates zone list incompatibility; this message is sent out periodically until the condition is corrected or **debug apple-errors** is turned off.

Most of the other messages that **debug apple-errors** can generate are quite obscure or indicate a very serious problem with the AppleTalk network. Some of these other messages follow.

In the following message, RTMPReq, RTMPResp, ATP, AEP, ZIP, ADSP, or SNMP could replace NBP, and llap dest not for us could replace wrong encapsulation.

```
Packet discarded, src 4160.12-254,dst 4160.19-254,NBP,wrong encapsulation
```

In the following message, besides invalid echo packet, other possible errors are: unsolicited AEP echo reply, unknown echo function, invalid ping packet, unknown ping function, and bad responder packet type.

```
Ethernet0: AppleTalk packet error; no source address available
AT: pak_reply: dubious reply creation, dst 4160.19
AT: Unable to get a buffer for reply to 4160.19
```

```
Processing error, src 4160.12-254,dst 4160.19-254,AEP, invalid echo packet
```

The **debug apple-errors** command can print out other messages when other debugging commands are also turned on. When both **debug apple-errors** and **debug apple-events** are turned on, the following message can be generated:

```
Proc err, src 4160.12-254,dst 4160.19-254,ZIP,NetInfo Reply format is invalid
```

In the previous message, besides NetInfo Reply format is invalid, other possible errors are: NetInfoReply not for me, NetInfoReply ignored, NetInfoReply for operational net ignored, NetInfoReply from invalid port, unexpected NetInfoReply ignored, cannot establish primary zone, no primary has been set up, primary zone invalid, net information mismatch, multicast mismatch, and zones disagree.

When both **debug apple-errors** and **debug apple-nbp** are turned on, the following message can be generated:

```
Processing error, ...,NBP,NBP name invalid
```

In the previous message, besides NBP name invalid, other possible errors are: NBP type invalid, NBP zone invalid, not operational, error handling brq, error handling proxy, NBP fwdreq unexpected, No route to srcnet, Proxy to "*" zone, Zone "*" from extended net, No zone info for "*", and NBP zone unknown.

When both **debug apple-errors** and **debug apple-routing** are turned on, the following message can be generated:

```
Processing error, ...,RTMPReq, unknown RTMP request
```

In the previous message, besides unknown RTMP request, other possible errors are: RTMP packet header bad, RTMP cable mismatch, routed RTMP data, RTMP bad tuple, and Not Req or Rsp.

Debug Apple-Events

Use the **debug apple-events** command to display debugging information about AppleTalk special events, neighbors becoming reachable/unreachable, and interfaces going up/down. Only significant events (for example, neighbor and/or route changes) are logged.

The **debug apple-events** command is very useful for solving AppleTalk network problems, because it provides an overall picture of the stability of the network. In a stable network, the **debug apple-events** command does not return any information. If, however, the command generates numerous messages, these messages can indicate where the problem might lie.

When configuring or making changes to a router or interface for AppleTalk, enable **debug apple-events**. This will alert you to the progress of the changes or to any errors that might result. You should also use this command periodically when you suspect network problems.

The **debug apple-events** command is also useful to determine whether network flapping is occurring. If flapping (nodes toggling on- and off-line) is excessive, look for routers that only support 254 networks.

When you enable **debug apple-events**, you also will see any messages that the configuration command **apple event-logging** normally displays. Turning on **debug apple-events**, however, will not cause **apple event-logging** to be maintained in nonvolatile memory. Only turning on **apple event-logging** explicitly will store it in nonvolatile memory. Furthermore, if **apple event-logging** is already enabled, turning on or off **debug apple-events** will not affect **apple event-logging**.

Figure 10-4 shows example **debug apple-events** output that describes a nonseed router coming up in discovery mode.

Discovery
mode state
change

```
Ether0: AT: Resetting interface address filters
%AT-5-INTRESTART: Ether0: AppleTalk port restarting; protocol restarted
Ether0: AppleTalk state changed; unknown -> restarting
Ether0: AppleTalk state changed; restarting -> probing
%AT-6-ADDRUSED: Ether0: AppleTalk node up; using address 65401.148
Ether0: AppleTalk state changed; probing -> acquiring
%AT-6-ACQUIREMODE: Ether0: AT port initializing; acquiring net configuration
Ether0: AppleTalk state changed; acquiring -> restarting
Ether0: AppleTalk state changed; restarting -> line down
Ether0: AppleTalk state changed; line down -> restarting
Ether0: AppleTalk state changed; restarting -> probing
%AT-6-ADDRUSED: Ether0: AppleTalk node up; using address 4160.148
Ether0: AppleTalk state changed; probing -> acquiring
%AT-6-ACQUIREMODE: Ether0: AT port initializing; acquiring net configuration
Ether0: AppleTalk state changed; acquiring -> requesting zones
Ether0: AT: Resetting interface address filters
%AT-5-INTRESTART: Ether0: AppleTalk port restarting; protocol restarted
Ether0: AppleTalk state changed; requesting zones -> verifying
AT: Sent GetNetInfo request broadcast on Ethernet0
Ethernet0: AppleTalk state changed; verifying -> checking zones
Ethernet0: AppleTalk state changed; checking zones -> operational
```

Figure 10-4 Example Debug Apple-Events Output with Discovery Mode State Changes

As Figure 10-4 shows, the **debug apple-events** command can be very useful in tracking the discovery mode state changes through which an interface progresses. When no problems are encountered, the state changes progress as follows:

1. Line down
2. Restarting
3. Probing (for its own address (node ID) using AARP)
4. Acquiring (sending out GetNetInfo requests)
5. Requesting zones (the list of zones for its cable)
6. Verifying (that the router's configuration is correct. If not, a port configuration mismatch is declared.)
7. Checking zones (to make sure its list of zones is correct)
8. Operational (participating in routing)

Explanations for individual lines of output in Figure 10-4 follow.

The following message indicates that a port is set. In this case, the zone multicast address is being reset.

```
Ether0: AT: Resetting interface address filters
```

The following messages indicate that the router is changing to restarting mode.

```
%AT-5-INTRESTART: Ether0: AppleTalk port restarting; protocol restarted  
Ether0: AppleTalk state changed; unknown -> restarting
```

The following message indicates that the router is probing in the startup range of network numbers (65280-65534) to discover its network number.

```
Ether0: AppleTalk state changed; restarting -> probing
```

The following message indicates that the router is enabled as a nonrouting node using a provisional network number within its startup range of network numbers. This type of message only appears if the network address the router will use differs from its configured address. This is always the case for a nonseed router; it is rarely the case for a seed router.

```
%AT-6-ADDRUSED: Ether0: AppleTalk node up; using address 65401.148
```

The following messages indicate that the router is sending out GetNetInfo requests to discover the default zone name and the actual network number range in which its network number should be chosen.

```
Ether0: AppleTalk state changed; probing -> acquiring  
%AT-6-ACQUIREMODE: Ether0: AT port initializing; acquiring net configuration
```

Now that the router has acquired the cable configuration information, the following message indicates that it restarts using that information.

```
Ether0: AppleTalk state changed; acquiring -> restarting
```

The following messages indicate that the router is probing for its actual network address.

```
Ether0: AppleTalk state changed; restarting -> line down  
Ether0: AppleTalk state changed; line down -> restarting  
Ether0: AppleTalk state changed; restarting -> probing
```

The following message indicates that the router has found an actual network address to use.

```
%AT-6-ADDRUSED: Ether0: AppleTalk node up; using address 4160.148
```

The following messages indicate that the router is sending out GetNetInfo requests to verify the default zone name and the actual network number range from which its network number should be chosen.

```
Ether0: AppleTalk state changed; probing -> acquiring
%AT-6-ACQUIREMODE: Ether0: AT port initializing; acquiring net configuration
```

The following message indicates that the router is requesting the list of zones for its cable.

```
Ether0: AppleTalk state changed; acquiring -> requesting zones
```

The following messages indicate that the router is sending out GetNetInfo requests to make sure its understanding of the configuration is correct.

```
Ether0: AppleTalk state changed; requesting zones -> verifying
AT: Sent GetNetInfo request broadcast on Ethernet0
```

The following message indicates that the router is rechecking its list of zones for its cable.

```
Ethernet0: AppleTalk state changed; verifying -> checking zones
```

The following message indicates that the router is now fully operational as a routing node and can begin routing.

```
Ethernet0: AppleTalk state changed; checking zones -> operational
```

Figure 10-5 shows example **debug apple-events** output that describes a seed router coming up when no other router is on the wire.

```
Ethernet1: AT: Resetting interface address filters
%AT-5-INTRESTART: Ethernet1: AppleTalk port restarting; protocol restarted
Ethernet1: AppleTalk state changed; unknown -> restarting
Ethernet1: AppleTalk state changed; restarting -> probing
%AT-6-ADDRUSED: Ethernet1: AppleTalk node up; using address 4165.204
Ethernet1: AppleTalk state changed; probing -> verifying
AT: Sent GetNetInfo request broadcast on Ethernet1
Ethernet1: AppleTalk state changed; verifying -> operational
%AT-6-ONLYROUTER: Ethernet1: AppleTalk port enabled; no neighbors found
```

Figure 10-5 Example Debug Apple-Events Output Showing Seed Coming Up by Itself

As Figure 10-5 shows, a seed router can come up when no other router is on the wire; however, it must assume that its configuration (if accurate syntactically) is correct, because no other router can verify it. Notice that the last line in Figure 10-5 indicates this situation.

Figure 10-6 shows example **debug apple-events** output that describes a nonseed router coming up when there is no seed router on the wire.

```

Ether0: AT: Resetting interface address filters
%AT-5-INTRESTART: Ether0: AppleTalk port restarting; protocol restarted
Ether0: AppleTalk state changed; unknown -> restarting
Ether0: AppleTalk state changed; restarting -> probing
%AT-6-ADDRUSED: Ether0: AppleTalk node up; using address 65401.148
Ether0: AppleTalk state changed; probing -> acquiring
AT: Sent GetNetInfo request broadcast on Ether0
AT: Sent GetNetInfo request broadcast on Ether0
AT: Sent GetNetInfo request broadcast on Ether0
AT: Sent GetNetInfo request broadcast on Ether0
AT: Sent GetNetInfo request broadcast on Ether0

```

Figure 10-6 Example Debug Apple-Events Output Showing NonSeed with No Seed

As Figure 10-6 shows, when you attempt to bring up a nonseed router without a seed router on the wire, it never becomes operational; instead, it hangs in the acquiring mode and continues to send out periodic GetNetInfo requests.

Figure 10-7 shows example **debug apple-events** output when a seed router is brought up on an AppleTalk internet that is in compatibility mode (set up to accommodate extended as well as nonextended AppleTalk) and the router has violated internet compatibility.

**Indicates
configuration
mismatch**



```

E0: AT: Resetting interface address filters
%AT-5-INTRESTART: E0: AppleTalk port restarting; protocol restarted
E0: AppleTalk state changed; restarting -> probing
%AT-6-ADDRUSED: E0: AppleTalk node up; using address 41.19
E0: AppleTalk state changed; probing -> verifying
AT: Sent GetNetInfo request broadcast on Ethernet0
%AT-3-ZONEDISAGREES: E0: AT port disabled; zone list incompatible with 41.19
AT: Config error for E0, primary zone invalid
E0: AppleTalk state changed; verifying -> config mismatch

```

Figure 10-7 Example Debug Apple-Events Output Showing Compatibility Conflict

The three configuration command lines that follow indicate the part of the router's configuration that caused the configuration mismatch shown in Figure 10-7.

```

lestat(config)#int e 0
lestat(config-if)#apple cab 41-41
lestat(config-if)#apple zone Marketign

```

The router shown in Figure 10-7 had been configured with a cable range of 41-41 instead of 40-40, which would have been accurate. To make matters worse, the zone name was configured incorrectly; the zone name should have been Marketing, rather than being misspelled as Marketign.

Debug Apple-NBP

Use the **debug apple-nbp** command to enable debugging output from the Name Binding Protocol (NBP) routines. To determine whether the router is receiving NBP lookups from a node on the AppleTalk network, enable **debug apple-nbp** at each node between the router and this node in order to determine where the problem lies.

Note: Because the **debug apple-nbp** command can generate a lot of messages, you should only use it when the router's CPU utilization is less than 50 percent.

Figure 10-8 shows example **debug apple-nbp** output.

```
AT: NBP ctrl = LkUp, ntuples = 1, id = 77
AT: 4160.19, skt 2, enum 0, name: =:ciscoRouter@Low End SW Lab
AT: LkUp =:ciscoRouter@Low End SW Lab

AT: NBP ctrl = LkUp-Reply, ntuples = 1, id = 77
AT: 4160.154, skt 254, enum 1, name: leststat.Ether0:ciscoRouter@Low End SW Lab

AT: NBP ctrl = LkUp, ntuples = 1, id = 78
AT: 4160.19, skt 2, enum 0, name: =:IPADDRESS@Low End SW Lab
AT: NBP ctrl = LkUp, ntuples = 1, id = 79
AT: 4160.19, skt 2, enum 0, name: =:IPGATEWAY@Low End SW Lab
AT: NBP ctrl = LkUp, ntuples = 1, id = 83
AT: 4160.19, skt 2, enum 0, name: =:ciscoRouter@Low End SW Lab
AT: LkUp =:ciscoRouter@Low End SW Lab

AT: NBP ctrl = LkUp, ntuples = 1, id = 84
AT: 4160.19, skt 2, enum 0, name: =:IPADDRESS@Low End SW Lab

AT: NBP ctrl = LkUp, ntuples = 1, id = 85
AT: 4160.19, skt 2, enum 0, name: =:IPGATEWAY@Low End SW Lab
AT: NBP ctrl = LkUp, ntuples = 1, id = 85
AT: 4160.19, skt 2, enum 0, name: =:IPGATEWAY@Low End SW Lab
```

Figure 10-8 Example Debug Apple-NBP Output

The first three lines in Figure 10-8 describe an NBP lookup request.

```
AT: NBP ctrl = LkUp, ntuples = 1, id = 77
AT: 4160.19, skt 2, enum 0, name: =:ciscoRouter@Low End SW Lab
AT: LkUp =:ciscoRouter@Low End SW Lab
```

Table 10-2 describes the fields in the first line of output in Figure 10-8.

Table 10-2 Debug Apple-NBP Field Descriptions—Part 1

Field	Description
AT: NBP	Indicates that this message describes an AppleTalk NBP packet.
ctrl = LkUp	Identifies the type of NBP packet. Possible values include: <ul style="list-style-type: none"> ■ LkUp—NBP lookup request. ■ LkUp-Reply—NBP lookup reply.
ntuples = 1	Indicates the number of name-address pairs in the lookup request packet. Range: 1-31 tuples.
id = 77	Value that identifies the NBP lookup request.

Table 10-3 describes the fields in the second line of output in Figure 10-8.

Table 10-3 Debug Apple-NBP Field Descriptions—Part 2

Field	Description
AT:	Indicates that this message describes an AppleTalk packet.
4160.19	Network address of the requester.
skt 2	Internet socket address of the requester. The responder will send the NBP lookup reply to this socket address.
enum 0	Enumerator field. Used to identify multiple names registered on a single socket. Each tuple is assigned its own enumerator, incrementing from 0 for the first tuple.
name: =:ciscoRouter@- Low End SW Lab	Entity name for which a network address has been requested. The AppleTalk entity name includes three components: <ul style="list-style-type: none"> ■ Object (in this case, the object is a wildcard character (=), indicating that the requester is requesting name-address pairs for all objects of the specified type in the specified zone) ■ Type (in this case, the type is ciscoRouter) ■ Zone (in this case, the zone is Low End SW Lab)

The third line in Figure 10-8 essentially reiterates the information in the two lines above it, indicating that a Lookup request has been made regarding name-address pairs for all objects of the ciscoRouter type in the Low End SW Lab zone.

Since the router is defined as an object of type `ciscoRouter` in zone `Low End SW Lab`, the router sends an NBP lookup reply in response to this NBP lookup request. The following two lines of output from Figure 10-8 show the router's response.

```
AT: NBP ctrl = LkUp-Reply, ntuples = 1, id = 77
AT: 4160.154, skt 254, enum 1, name: lestat.Ether0:ciscoRouter@Low End SW Lab
```

In the first line, `ctrl = LkUp-Reply` identifies this NBP packet as an NBP lookup request. The same value in the `id` field (`id = 77`) associates this lookup reply with the previous lookup request. The second line indicates that the network address associated with the router's entity name (`lestat.Ether0:ciscoRouter@Low End SW Lab`) is `4160.154`. The fact that no other entity name/network address is listed indicates that the responder only knows about itself as an object of type `ciscoRouter` in zone `Low End SW Lab`.

Debug Apple-Packet

Use the **debug apple-packet** command to enable per-packet debugging output. It reports information on line when a packet is received or a transmit is attempted. The command allows you to monitor the types of packets being slow switched. It is roughly equivalent to turning on all the other AppleTalk debugging information. There will be at least one line of debugging output per AppleTalk packet processed.

When invoked in conjunction with the commands **debug apple-routing**, **debug apple-zip**, and **debug apple-nbp**, the **debug apple-packet** command adds protocol processing information in addition to generic packet details. It reports protocol processing and successful completion or failure information.

When invoked in conjunction with the command **debug apple-errors**, the **debug apple-packet** command reports packet-level problems, such as those concerning encapsulation, for example.

Note: Because the **debug apple-packet** command can generate a lot of messages, you should only use it when the router's CPU utilization is less than 50 percent.

Figure 10-9 shows example **debug apple-packet** output.

```
Ether0: AppleTalk packet: enctype SNAP, size 60, encaps000000000000000000000000
AT: src=Ethernet0:4160.47, dst=4160-4160, size=10, 2 rtes, RTMP pkt sent
AT: ZIP Extended reply rcvd from 4160.19
AT: ZIP Extended reply rcvd from 4160.19
AT: src=Ethernet0:4160.47, dst=4160-4160, size=10, 2 rtes, RTMP pkt sent
Ether0: AppleTalk packet: enctype SNAP, size 60, encaps000000000000000000000000
Ether0: AppleTalk packet: enctype SNAP, size 60, encaps000000000000000000000000
```

Figure 10-9 Example Debug Apple-Packet Output

Table 10-4 describes the fields in the first line of output shown in Figure 10-9.

Table 10-4 Debug Apple-Packet Field Descriptions—Part 1

Field	Description
Ether0:	Name of the interface through which the router received the packet.
AppleTalk packet	Indicates that this is an AppleTalk packet.
enctype SNAP	Encapsulation type for the packet.
size 60	Size of the packet (in bytes).
encaps000000000000000000000000	Encapsulation.

Table 10-5 describes the fields in the second line of output shown in Figure 10-9.

Table 10-5 Debug Apple-Packet Field Descriptions—Part 2

Field	Description
AT:	Indicates that this is an AppleTalk packet.
src = Ethernet0:4160.47	Name of the interface sending the packet, as well as its AppleTalk address.
dst = 4160-4160	Cable range of the packet's destination.
size = 10	Size of the packet (in bytes).
2 rtes	Indicates that there are two routes in the routing table that link these two addresses.
RTMP pkt sent	Indicates the type of packet sent.

The third line in Figure 10-9 indicates the type of packet received and its source AppleTalk address. This message is repeated in the fourth line because AppleTalk hosts can send multiple replies to a given GetNetInfo request.

Debug Apple-Routing

Use the **debug apple-routing** command to enable debugging output from the Routing Table Maintenance Protocol (RTMP) routines. This command can be used to monitor acquisition of routes, aging of routing table entries, and advertisement of known routes. It also reports conflicting network numbers on the same network if the network is misconfigured.

Note: Because the **debug apple-routing** command can generate a lot of messages, you should only use it when the router's CPU utilization is less than 50 percent.

Figure 10-10 shows example **debug apple-routing** output.

```
AT: src=Ethernet0:4160.41, dst=4160-4160, size=19, 2 rtes, RTMP pkt sent
AT: src=Ethernet1:41069.25, dst=41069, size=427, 96 rtes, RTMP pkt sent
AT: src=Ethernet2:4161.23, dst=4161-4161, size=427, 96 rtes, RTMP pkt sent
AT: Route ager starting (97 routes)
AT: Route ager finished (97 routes)
AT: RTMP from 4160.19 (new 0,old 94,bad 0,ign 0, dwn 0)
AT: RTMP from 4160.250 (new 0,old 0,bad 0,ign 2, dwn 0)
AT: RTMP from 4161.236 (new 0,old 94,bad 0,ign 1, dwn 0)
AT: src=Ethernet0:4160.41, dst=4160-4160, size=19, 2 rtes, RTMP pkt sent
```

Figure 10-10 Example Debug Apple-Routing Output

Explanations for representative lines of the **debug apple-routing** output in Figure 10-10 follow.

Table 10-6 describes the fields in the first line of example **debug apple-routing** output.

Table 10-6 Debug Apple-Routing Field Descriptions—Part 1

Field	Description
AT:	Indicates that this is AppleTalk debugging output.
src = Ethernet0:4160.41	Indicates the source router interface and network address for the RTMP update packet.
dst = 4160-4160	Indicates the destination network address for the RTMP update packet.
size = 19	Size of this RTMP packet (in bytes).
2 rtes	This RTMP update packet includes information on two routes.
RTMP pkt sent	Indicates that this type of message describes an RTMP update packet that the router has sent (rather than one that it has received).

The following two messages indicate that the ager has started and finished the aging process for the routing table and that this table contains 97 entries.

```
AT: Route ager starting (97 routes)
AT: Route ager finished (97 routes)
```

Table 10-7 describes the fields in the following line of **debug apple-routing** output.

```
AT: RTMP from 4160.19 (new 0,old 94,bad 0,ign 0, dwn 0)
```

Table 10-7 Debug Apple-Routing Field Descriptions—Part 2

Field	Description
AT:	Indicates that this is AppleTalk debugging output.
RTMP from 4160.19	Indicates the source address of the RTMP update the router received.
new 0	Indicates the number of routes in this RTMP update packet that the router did not already know about.
old 94	Indicates the number of routes in this RTMP update packet that the router already knew about.
bad 0	Number of routes the other router indicates have gone bad.
ign 0	Number of routes the other router indicates it does not care about.
dwn 0	Number of poisoned tuples included in this packet.

Debug Apple-ZIP

Use the **debug apple-zip** command to enable debugging output from the Zone Information Protocol (ZIP) routines. This command reports significant events such as discovery of new zones and zone list queries. It generates information similar to what **debug apple-routing** generates, but for ZIP packets instead of RTMP packets.

The **debug apple-zip** command can be used to determine whether a ZIP storm is taking place in the AppleTalk network. You can detect the existence of a ZIP storm when you see that no router on a cable has the zone name corresponding to a network number that all the routers have in their routing tables.

Figure 10-11 shows example **debug apple-zip** output.

```
AT: Sent GetNetInfo request broadcast on Ether0
AT: Recvd ZIP cmd 6 from 4160.19-6
AT: 3 query packets sent to neighbor 4160.19
AT: 1 zones for 31902, ZIP XReply, src 4160.19
AT: net 31902, zonelen 10, name US-Orlando
```

Figure 10-11 Example Debug Apple-ZIP Output

Explanations of the lines of output shown in Figure 10-11 follow.

The first line indicates that the router has received an RTMP update that includes a new network number and is now requesting zone information.

```
AT: Sent GetNetInfo request broadcast on Ether0
```

The second line indicates that the neighbor at address 4160.19 replies to the zone request with a default zone.

```
AT: Recvd ZIP cmd 6 from 4160.19-6
```

In response, the third line shows that the router sends three queries to the neighbor at network address 4160.19 for other zones on the network.

```
AT: 3 query packets sent to neighbor 4160.19
```

The fourth line suggests that the neighbor at network address 4160.19 responds with a ZIP extended reply, indicating that one zone has been assigned to network 31902.

```
AT: 1 zones for 31902, ZIP XReply, src 4160.19
```

The fifth line indicates that the router responds that the zone name of network 31902 is US-Orlando, and the zone length of that zone name is 10.

```
AT: net 31902, zonelen 10, name US-Orlando
```

Debug ARP

Use the **debug arp** command to display information on ARP protocol transactions.

Use this command when some nodes on a TCP/IP network are responding, but others are not. It shows whether or not the router is sending or receiving ARPs.

Figure 10-12 shows example **debug arp** output.

```
IP ARP: sent req src 131.108.22.7 0000.0c01.e117, dst 131.108.22.96 0000.0000.0000
IP ARP: rcvd rep src 131.108.22.96 0800.2010.b908, dst 131.108.22.7
IP ARP: rcvd req src 131.108.6.10 0000.0c00.6fa2, dst 131.108.6.62
IP ARP: rep filtered src 131.108.22.7 aa92.1b36.a456, dst 255.255.255.255 ffff.ffff.ffff
IP ARP: rep filtered src 131.108.9.7 0000.0c00.6b31, dst 131.108.22.7 0800.2010.b908
```

Figure 10-12 Example Debug ARP Output

In Figure 10-12, each line of output represents an ARP packet that the router sent or received. Explanations for the individual lines of output follow.

The first line indicates that the router at IP address 131.108.22.7 and MAC address 0000.0c01.e117 sent an ARP request for the MAC address of the host at 131.108.22.96. The series of zeros (0000.0000.0000) following this address indicate that the router is currently unaware of the MAC address.

```
IP ARP: sent req src 131.108.22.7 0000.0c01.e117, dst 131.108.22.96 \
0000.0000.0000
```

The second line indicates that the router at IP address 131.108.22.7 receives a reply from the host at 131.108.22.96 indicating that its MAC address is 0800.2010.b908.

```
IP ARP: rcvd rep src 131.108.22.96 0800.2010.b908, dst 131.108.22.7
```

The third line indicates that the router receives an ARP request from the host at 131.108.6.10 requesting the MAC address for the host at 131.108.6.62.

```
IP ARP: rcvd req src 131.108.6.10 0000.0c00.6fa2, dst 131.108.6.62
```

The fourth line indicates that another host on the network attempted to send the router an ARP reply for the router's own address. The router ignores such bogus replies. This usually can happen if someone is running a bridge in parallel with the router and is allowing ARP to be bridged. It indicates a network misconfiguration.

```
IP ARP: rep filtered src 131.108.22.7 aa92.1b36.a456, dst 255.255.255.255 \
ffff.ffff.ffff
```

The fifth line indicates that another host on the network attempted to inform the router that it is on network 131.108.9.0, but the router does not know that that network is attached to a different router interface. The remote host (probably a PC or an X terminal) is misconfigured. If the router were to install this entry, it would deny service to the real machine on the proper cable.

```
IP ARP: rep filtered src 131.108.9.7 0000.0c00.6b31, dst 131.108.22.7 \
0800.2010.b908
```


Debug Broadcast

Use the **debug broadcast** command to debug MAC broadcast packets.

Depending on the type of interface and the type of encapsulation used on that interface, the **debug broadcast** command can produce a wide range of messages.

Figure 10-13 shows example **debug broadcast** output. Notice how similar it is to the **debug packet** output shown in Figure 10-34.

```
Ethernet0: Broadcast ARPA, src 0000.0c00.6fa4, dst ffff.ffff.ffff, type 0x0800,
data 4500002800000000FF11EA7B, len 60
Serial3: Broadcast HDLC, size 64, type 0x800, flags 0x8F00
Serial2: Broadcast PPP, size 128
Serial7: Broadcast FRAME-RELAY, size 174, type 0x800, DLCI 7a
Ultraset0: Broadcast ULTRANET, size 60
```

Figure 10-13 Example Debug Broadcast Output

Table 10-8 describes significant fields shown in Figure 10-13.

Table 10-8 Debug Broadcast Field Descriptions

Field	Description
Ethernet0	Name of Ethernet interface that received the packet.
Broadcast	States that this packet was a broadcast packet.
ARPA	States that this packet uses ARPA-style encapsulation. Possible encapsulation styles vary depending on the media, as follows.
	<i>Media Type</i> <i>Encapsulation Style</i>
	Ethernet APOLLO ARP ETHERTALK ISO1 ISO3 LLC2 NOVELL-ETHER SNAP
	FDDI APOLLO ISO1 ISO3 LLC2 SNAP

Field	Description	
ARPA (continued)	Serial	BFEX25 BRIDGE DDN-X25 DDNX25-DCE ETHERTALK FRAME-RELAY HDLC HDH LAPB LAPBDCE MULTI-LAPB PPP SDLC-PRIMARY SDLC-SECONDARY SLIP SMDS STUN X25 X25-DCE
	Token Ring	3COM-TR ISO1 ISO3 MAC LLC2 NOVELL-TR SNAP VINES-TR
	Ultramet	ULTRANET ULTRANET-HELLO
src 0000.0c00.6fa4	MAC address of the node generating the packet.	
dst ffff.ffff.ffff.ffff	MAC address of the destination node for the packet. This address is always the MAC broadcast address.	
type 0x0800	Packet type (IP in this case).	
data ...	First 12 bytes of the datagram following the MAC header.	
len 60	Length of the message that the interface received from the wire (in bytes).	
size 128	Length of the message that the interface received from the wire (in bytes).	
flags 0x8F00	HDLC or PPP flags field.	
DLCI 7a	The DLCI number on Frame Relay.	

Debug DECnet-Connects

Use the **debug decnet-connects** command to enable logging of all connect packets that are filtered (permitted or denied) by DECnet access lists.

When using connect packet filtering, it may be helpful to use the **decnet access-group** configuration command to apply the following basic access list:

```
access-list 300 permit 0.0 63.1023
access-list 300 permit 0.0 63.1023 eq any
```

You can then log all connect packets transmitted on interfaces to which you applied this list, in order to determine those elements on which your connect packets must be filtered.

Figure 10-14 shows example **debug decnet-connect** output.

```
DNET: list 300 item #2 matched src=19.403 dst=19.309 on Ethernet0: permitted
      srcname="RICK" srcuic=[0,017]
      dstobj=42 id="USER"
```

Figure 10-14 Example Debug DECnet-Connect Output

Table 10-9 describes significant fields shown in Figure 10-14.

Table 10-9 Debug DECnet-Connects Field Descriptions

Field	Description
DNET:	Indicates that this is a DECnet packet.
list 300 item #2 matched	Indicates that a packet matched the second item in access list 300.
src = 19.403	Indicates the source DECnet address for the packet.
dst = 19.309	Indicates the destination DECnet address for the packet.
on Ethernet0:	Indicates the router interface on which the access list filtering the packet was applied.
permitted	Indicates that the access list permitted the packet.
srcname = "RICK"	Indicates the originator user of the packet.
srcuic = [0,017]	Indicates the source UIC of the packet.
dstobj = 42	Indicates that DECnet object 42 is the destination.
ID = "USER"	Indicates the access user.

Note: Packet password and account information is not logged in the **debug decnet-connects** message, nor is it displayed by the **show access EXEC** command. If you specify **password** or **account** information in your access list, they will be viewable by anyone with access to your router's configuration.

Debug Frame-Relay

Use the **debug frame-relay** command to analyze the packets that have been received on a frame relay interface. (To analyze the packets that have been *sent* on a frame relay interface, use the **debug frame-relay-packets** command.)

Note: Because the **debug frame-relay** command generates a lot of output, you should only use it when traffic on the frame relay network is less than 25 packets per second.

Figure 10-15 shows example **debug frame-relay** output.

```
Serial0(i): dlci 500(0x7C41), pkt type 0x809B, datagramsize      24
Serial1(i): dlci 1023(0xFCF1), pkt type 0x309, datagramsize      13
Serial0(i): dlci 500(0x7C41), pkt type 0x809B, datagramsize      24
Serial1(i): dlci 1023(0xFCF1), pkt type 0x309, datagramsize      13
Serial0(i): dlci 500(0x7C41), pkt type 0x809B, datagramsize      24
```

Figure 10-15 Example Debug Frame-Relay-Packets Output

Table 10-10 describes significant fields shown in Figure 10-15.

Table 10-10 Debug Frame-Relay Field Descriptions

Field	Description
Serial0(i):	Indicates that the Serial0 interface has received this frame relay datagram as input.
dlci 500(0x7C41)	Value of the DLCI for this packet in decimal (and q922). In this case, 500 has been configured as the multicast DLCI.
pkt type 0x809B	Indicates the packet type code. Possible supported Ethernet type codes follow: 0x0200—PUP 0x0201—IP on 3MB net 0x0201—Xerox ARP on 10MB nets 0xCC—RFC 1294 (only for IP) 0x0600—XNS 0x0800—IP on 10MB net 0x0804—Chaos on 10MB net 0x0806—IP ARP 0x0808—Frame Relay ARP 0x0BAD—Vines IP 0x0BAE—Vines Loopback Protocol 0x0BAF—Vines Echo 0x6001—DEC MOP booting protocol 0x6002—DEC MOP console protocol 0x6003—DECnet Phase IV on Ethernet 0x6004—DEC LAT on Ethernet 0x8005—HP Probe 0x8035—RARP 0x8038—DEC spanning tree 0x809b—Apple EtherTalk 0x80f3—AppleTalk ARP 0x8019—Apollo domain 0x80C4—VINES IP 0x80C5—VINES ECHO 0x8137—Novell IPX 0x9000—Ethernet loopback packet Possible HDLC type codes follow: 0x1A58—Novell IPX, standard form 0xFEFE—CLNS 0xEFEF—ES-IS 0x1998—Uncompressed TCP 0x1999—Compressed TCP 0x6558—Serial line bridging
datagramsize 24	Size of this datagram (in bytes)

Debug Frame-Relay-Events

Use the **debug frame-relay-events** command to display information about frame relay ARP replies on networks that support a multicast channel and use dynamic addressing.

This command is most useful for identifying the cause of end-to-end connection problems during the installation of a frame relay network or node.

Note: Because the **debug frame-relay-events** command does not generate a lot of output, you can use it even during production hours.

Figure 10-16 shows example **debug frame-relay-events** output.

```
Serial2(i): reply rcvd 131.108.170.26 126
Serial2(i): reply rcvd 131.108.170.28 128
Serial2(i): reply rcvd 131.108.170.34 134
Serial2(i): reply rcvd 131.108.170.38 144
Serial2(i): reply rcvd 131.108.170.41 228
Serial2(i): reply rcvd 131.108.170.65 325
```

Figure 10-16 Example Debug Frame-Relay-Events Output

As Figure 10-16 suggests, **frame-relay-events** returns one specific message type. The first line, for example, indicates that IP address 131.108.170.26 sent a frame relay ARP reply; this packet was received as input on the Serial2 interface. The last field (126) is the DLCI to use when communicating with the responding router.

Debug Frame-Relay-LMI

Use the **debug frame-relay-lmi** command to display information on the local management interface (LMI) packets exchanged by the router and the frame relay service provider.

You can use this command to determine whether the router and the frame relay switch are sending and receiving LMI packets properly.

Note: Because the **debug frame-relay-lmi** command does not generate much output, you can use it even during periods of heavy traffic.

Figure 10-17 shows example **debug frame-relay-lmi** output.

The diagram illustrates the output of the **debug frame-relay-lmi** command. It is divided into two sections by arrows pointing to specific lines of output. The first section, labeled "LMI exchange", points to the first three lines of output, which describe an LMI request and its response. The second section, labeled "Full LMI status message", points to the last five lines of output, which describe the router's two PVCs.

```
Serial1(out): clock 20212760, myseq 206, mineseen 205, yourseen 136, line up
RT IE 1, length 1, type 1
KA IE 3, length 2, yourseq 138, myseq 206
Serial1(out): clock 20222760, myseq 207, mineseen 206, yourseen 138, line up
RT IE 1, length 1, type 1
KA IE 3, length 2, yourseq 140, myseq 207
Serial1(out): clock 20232760, myseq 208, mineseen 207, yourseen 140, line up
RT IE 1, length 1, type 1
KA IE 3, length 2, yourseq 142, myseq 208
Serial1(out): clock 20252760, myseq 210, mineseen 209, yourseen 144, line up
RT IE 1, length 1, type 0
KA IE 3, length 2, yourseq 146, myseq 210
PVC IE 0x7 , length 0x6 , dlci 400 , status 0 , bw 0x0 DAC0
PVC IE 0x7 , length 0x6 , dlci 401 , status 0 , bw 0x0 DAC0
```

Figure 10-17 Example Debug Frame-Relay-LMI Output

In Figure 10-17, the first three lines describe an LMI exchange. The first line describes the LMI request the router has sent to the switch. The second and third lines describe the response to this request from the switch. This LMI exchange is followed by two similar LMI exchanges. The last five lines in Figure 10-17 comprise a full LMI status message that includes a description of the router's two PVCs.

Table 10-11 describes significant fields in the first line of the **debug frame-relay-lmi** output shown in Figure 10-17.

Table 10-11 Debug Frame-Relay-LMI Field Descriptions—Part 1

Field	Description
Serial1(out)	Indicates that the LMI request was sent out on the Serial1 interface.
clock 20212760	System clock (in milliseconds). Useful for determining whether an appropriate amount of time has transpired between events.
myseq 206	The myseq counter maps to the router's CURRENT SEQ counter, as described in the Frame Relay Specification with Extensions.
mineseen 205	The mineseen counter maps to the router's LAST RCVD SEQ counter, as described in the Frame Relay Specification with Extensions.
yourseen 136	The yourseen counter maps to the LAST RCVD SEQ counter of the switch, as described in the Frame Relay Specification with Extensions.
line up	Indicates the line protocol up/down state.

Table 10-12 describes significant fields in the second and third lines of **debug frame-relay-lmi** output shown in Figure 10-17.

Table 10-12 Debug Frame-Relay-LMI Field Descriptions—Part 2

Field	Description
RT IE 1	Value of the report type information element.
length 1	Length of the Report Type Information Element (in bytes).
type 1	Report type in RT IE.
KA IE 3	Value of the keepalive information element.
length 2	Length of the Keep Alive Information Element (in bytes).
yourseq 138	The yourseq counter maps to the CURRENT SEQ counter of the switch, as described in the Frame Relay Specification with Extensions.
myseq 206	The myseq counter maps to the router's CURRENT SEQ counter, as described in the Frame Relay Specification with Extensions.

Table 10-13 describes significant fields in the last line of **debug frame-relay-lmi** output shown in Figure 10-17.

Table 10-13 Debug Frame-Relay-LMI Field Descriptions—Part 3

Field	Description
PVC IE 0x7	Value of the permanent virtual circuit information element type.
length 0x6	Length of the PVC IE (in bytes).
dlci 401	DLCI decimal value for this PVC.
status 0	Status value. Possible values include the following: <ul style="list-style-type: none">■ 0x00—Added/inactive■ 0x02—Added/active■ 0x04—Deleted■ 0x08—New/inactive■ 0x0a—New/active
bw 0x0 DAC0	CIR (committed information rate), in hex, for the DLCI.

Debug Frame-Relay-Packets

Use the **debug frame-relay-packets** command to analyze the packets that have been sent on a frame relay interface. (To analyze the packets that have been *received* on a frame relay interface, use the **debug frame-relay** command.)

Note: Because the **debug frame-relay-packets** command generates a lot of output, you should only use it when traffic on the frame relay network is less than 25 packets per second.

Figure 10-18 shows example **debug frame-relay-packets** output.

```
Serial0: broadcast = 1, link 809B, addr 65535.255
Serial0(o):DLCI 500 type 809B size 24
Serial0: broadcast = 0, link 809B, addr 10.2
Serial0(o):DLCI 100 type 809B size 104
Serial0: broadcast search
Serial0(o):DLCI 300 type 809B size 24
Serial0(o):DLCI 400 type 809B size 24
```

Figure 10-18 Example Debug Frame-Relay-Packets Output

As Figure 10-18 shows, **debug frame-relay-packets** output is made up of groups of output lines; each group describes a frame relay packet that has been sent. The number of lines in the group can vary, depending on the number of DLCIs on which the packet was sent. For example, the first two pairs of output lines describe two different packets, both of which were sent out on a single DLCI. The last three lines in Figure 10-18 describe a single frame relay packet that was sent out on two DLCIs.

Table 10-14 describes significant fields shown in the first pair of output lines in Figure 10-18.

Table 10-14 Debug Frame-Relay-Packets Field Descriptions

Field	Description
Serial0:	Indicates the interface that has sent the frame relay packet.
broadcast = 1	Indicates the destination of the packet. Possible values include the following: <ul style="list-style-type: none"> ■ broadcast = 1—Broadcast address ■ broadcast = 0—Particular destination ■ broadcast search—Searches all frame-relay map entries for this particular protocol that include the keyword broadcast.
link 809B	Indicates the packet type, as documented under Debug Frame-Relay.
addr 65535.255	Indicates the destination protocol address for this packet. In this case, it is an AppleTalk address.
Serial0(o):	(o) indicates that this is an output event.
DLCI 500	Decimal value of the DLCI.
type 809B	Indicates the packet type, as documented under Debug Frame-Relay.
size 24	Size of this packet (in bytes).

The discussion that follows describes the other lines of **debug frame-relay-packet** output shown in Figure 10-18.

The following group of output lines describes a frame relay packet sent to a particular address; in this case AppleTalk address 10.2.

```
Serial0: broadcast - 0, link 809B, addr 10.2
Serial0(o):DLCI 100 type 809B size 104
```

The following group of output lines describes a frame relay packet sent to a true broadcast address.

```
Serial1: broadcast search
Serial1(o):DLCI 400 type 800 size 288
```

The following group of output lines describes a frame relay packet that went out on two different DLCIs, because two frame relay map entries were found.

```
Serial0: broadcast search
Serial0(o):DLCI 300 type 809B size 24
Serial0(o):DLCI 400 type 809B size 24
```

Debug IP-ICMP

Use the **debug ip-icmp** command to enable logging of ICMP transactions.

This command is useful for determining whether the router is sending and/or receiving ICMP messages, when troubleshooting an end-to-end connection problem, for example.

Figure 10-19 shows example **debug ip-icmp** output.

```
ICMP: rcvd type 3, code 1, from 128.95.192.4
ICMP: src 36.56.0.202, dst 131.108.16.1, echo reply
ICMP: dst (131.120.1.0) port unreachable rcv from 131.120.1.15
ICMP: src 131.108.12.35, dst 131.108.20.7, echo reply
ICMP: dst (255.255.255.255) protocol unreachable rcv from 192.31.7.21
ICMP: dst (131.120.1.0) port unreachable rcv from 131.120.1.15
ICMP: dst (255.255.255.255) protocol unreachable rcv from 192.31.7.21
ICMP: dst (131.120.1.0) port unreachable rcv from 131.120.1.15
ICMP: src 36.56.0.202, dst 131.108.16.1, echo reply
ICMP: dst (131.120.1.0) port unreachable rcv from 131.120.1.15
ICMP: dst (255.255.255.255) protocol unreachable rcv from 192.31.7.21
ICMP: dst (131.120.1.0) port unreachable rcv from 131.120.1.15
```

Figure 10-19 Example Debug IP-ICMP Output

Table 10-15 describes significant fields shown in the first line of **debug ip-icmp** output shown in Figure 10-19.

Table 10-15 Debug IP-ICMP Field Descriptions—Part 1

Field	Description
ICMP:	Indicates that this message describes an ICMP packet.
rcvd type 3	<p>The type field can be one of the following:</p> <ul style="list-style-type: none"> ■ 0—Echo Reply ■ 3—Destination Unreachable ■ 4—Source Quench ■ 5—Redirect ■ 8—Echo ■ 9—Router Discovery Protocol Advertisement ■ 10—Router Discovery Protocol Solicitations ■ 11—Time Exceeded ■ 12—Parameter Problem ■ 13—Timestamp ■ 14—Timestamp Reply ■ 15—Information Request ■ 16—Information Reply ■ 17—Mask Request ■ 18—Mask Reply
code 1	<p>The next field is a code. The meaning of the code depends upon the type field value:</p> <p>Echo and Echo Reply—The code field is always zero.</p> <p>Destination Unreachable—The code field can have the following values:</p> <ul style="list-style-type: none"> 0—Network unreachable 1—Host unreachable 2—Protocol unreachable 3—Port unreachable 4—Fragmentation needed and DF bit set 5—Source route failed <p>Source Quench—The code field is always 0.</p> <p>Redirect—The code field can have the following values:</p> <ul style="list-style-type: none"> 0—Redirect datagrams for the Network 1—Redirect datagrams for the Host 2—Redirect datagrams for the Type of Service and Network 3—Redirect datagrams for the Type of Service and Host

Field	Description
code 1 (continued)	<p>Router Discovery Protocol Advertisements and Solicitations—The code field is always zero.</p> <p>Time Exceeded—The code field can have the following values:</p> <ul style="list-style-type: none"> 0—Time to live exceeded in transit 1—Fragment reassembly time exceeded <p>Parameter Problem—The code field can have the following values:</p> <ul style="list-style-type: none"> 0—General problem 1—Option is missing 2—Option missing, no room to add <p>Timestamp and Timestamp Reply—The code field is always zero.</p> <p>Information Request and Information Reply—The code field is always zero.</p> <p>Mask Request and Mask Reply—The code field is always zero.</p>
from 128.95.192.4	Indicates the source address of the ICMP packet.

Table 10-16 describes significant fields shown in the second line of **debug ip-icmp** output in Figure 10-19.

Table 10-16 Debug IP-ICMP Field Descriptions—Part 2

Field	Description
ICMP:	Indicates that this messages describes an ICMP packet.
src 36.56.0.202	The address of the sender of the echo.
dst 131.108.16.1	The address of the receiving router.
echo reply	Indicates the router received an echo reply.

Other messages that the **debug ip-icmp** command can generate follow.

When an IP router or host sends out an ICMP mask request, the following message is generated when the router sends a mask reply.

```
ICMP: sending mask reply (255.255.255.0) to 160.89.80.23 via Ethernet0
```

The following two lines are examples of the two forms of this message. The first form is generated when a mask reply comes in after the router sends out a mask request. The second form occurs when the router receives a mask reply with a nonmatching sequence and ID. See Appendix I of RFC 950 for details.

```
ICMP: mask reply 255.255.255.0 from 160.89.80.31
ICMP: unexpected mask reply 255.255.255.0 from 160.89.80.32
```

The following output indicates that the router sent a redirect packet to the host at address 160.89.80.31, instructing that host to use the gateway at address 160.89.80.23 in order to reach the host at destination address 131.108.1.111.

```
ICMP: redirect sent to 160.89.80.31 for dest 131.108.1.111 use gw 160.89.80.23
```

The following message indicates that the router received a redirect packet from the host at address 160.89.80.23, instructing the router to use the gateway at address 160.89.80.28 in order to reach the host at destination address 160.89.81.34.

```
ICMP: redirect rcvd from 160.89.80.23 -- for 160.89.81.34 use gw 160.89.80.28
```

The following message is displayed when the router sends an ICMP packet to the source address (160.89.94.31 in this case) indicating that the destination address (131.108.13.33 in this case) is unreachable.

```
ICMP: dst (131.108.13.33) host unreachable sent to 160.89.94.31
```

The following message is displayed when the router receives an ICMP packet from an intermediate address (160.89.98.32 in this case) indicating that the destination address (131.108.13.33 in this case) is unreachable.

```
ICMP: dst (131.108.13.33) host unreachable rcv from 160.89.98.32
```

Depending on the code received (as Table 10-15 describes), any of the unreachable messages can have any of the following instead of the “host” string in the message:

```
net
protocol
port
frag. needed and DF set
source route failed
prohibited
```

The following message is displayed when the TTL in the IP header reaches zero and a time exceed ICMP message is sent. The fields are self-explanatory.

```
ICMP: time exceeded (time to live) send to 128.95.1.4 (dest was 131.108.1.111)
```

The following message is generated when parameters in the IP header are corrupted in some way and the parameter problem icmp message is sent. Fields are self-explanatory.

```
ICMP: parameter problem sent to 128.121.1.50 (dest was 131.108.1.111)
```

Based on the preceding information, the remaining output should be easily understood.

```
ICMP: parameter problem rcvd 160.89.80.32
ICMP: source quench rcvd 160.89.80.32
ICMP: source quench sent to 128.121.1.50 (dest was 131.108.1.111)
ICMP: sending time stamp reply to 160.89.80.45
ICMP: sending info reply to 160.89.80.12
ICMP: rdp advert rcvd type 9, code 0, from 160.89.80.23
ICMP: rdp solicit rcvd type 10, code 0, from 160.89.80.43
```

Note: For more information about the fields in **debug ip-icmp** output, see RFC-792, “Internet Control Message Protocol;” Appendix I of RFC-950, “Internet Standard Subnetting Procedure;” and RFC-1256, “ICMP Router Discovery Messages.”

Debug IP-IGRP

Use the **debug ip-igrp** command to display IGRP routing transactions.

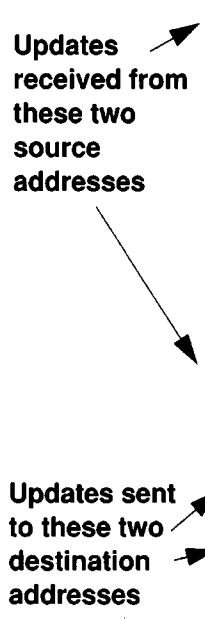
Syntax for the **debug ip-igrp** command follows.

debug ip-igrp [*ip-address*]

If the IP address of an IGRP neighbor is specified, the resulting **debug ip-igrp** output will include messages describing updates from that neighbor and updates that the router broadcasts toward that neighbor.

When there are many networks in your routing table, **debug ip-igrp** can flood the console and make the router unusable. In this case, use **debug ip-igrp-events** instead to display summary routing information.

Figure 10-20 shows example **debug ip-igrp** output.



Updates received from these two source addresses

```
IGRP: received update from 160.89.80.240 on Ethernet
subnet 160.89.66.0, metric 1300 (neighbor 1200)
subnet 160.89.56.0, metric 8676 (neighbor 8576)
subnet 160.89.48.0, metric 1200 (neighbor 1100)
subnet 160.89.50.0, metric 1300 (neighbor 1200)
subnet 160.89.40.0, metric 8676 (neighbor 8576)
network 192.82.152.0, metric 158550 (neighbor 158450)
network 192.68.151.0, metric 1115511 (neighbor 1115411)
network 150.136.0.0, metric 16777215 (inaccessible)
exterior network 129.140.0.0, metric 9676 (neighbor 9576)
exterior network 140.222.0.0, metric 9676 (neighbor 9576)
IGRP: received update from 160.89.80.28 on Ethernet
subnet 160.89.95.0, metric 180671 (neighbor 180571)
subnet 160.89.81.0, metric 1200 (neighbor 1100)
subnet 160.89.15.0, metric 16777215 (inaccessible)
IGRP: sending update to 255.255.255.255 via Ethernet0 (160.89.64.31)
subnet 160.89.94.0, metric=847
IGRP: sending update to 255.255.255.255 via Serial1 (160.89.94.31)
subnet 160.89.80.0, metric=16777215
subnet 160.89.64.0, metric=1100
```

Updates sent to these two destination addresses

Figure 10-20 Example Debug IP-IGRP Output

Figure 10-20 shows that the router being debugged has received updates from two other routers on the network. The router at source address 160.89.80.240 sent information about 10 destinations in the update; the router at source address 160.89.80.28 sent information about three destinations in its update. The router being debugged also sent updates—in both cases to the broadcast address 255.255.255.255 as the destination address.

The first line in Figure 10-20 is self explanatory.

On the second line in Figure 10-20, the first field refers to the type of destination information: “subnet” (interior), “network” (system) or “exterior” (exterior). The second field is the Internet address of the destination network. The third field is the metric stored in the routing table and the metric advertised by the neighbor sending the information. “Metric ... inaccessible” usually means that the neighbor router has put the destination in holddown.

The entries in Figure 10-20 showing that the router is sending updates are similar, except that the numbers in parentheses are the source addresses used in the IP header. A metric of 16777215 is inaccessible.

Other examples of output that the **debug ip-igrp** command can produce follow.

The following entry indicates that the routing table was updated and shows the new edition number (97 in this case) to be used in the next IGRP update.

```
IGRP: edition is now 97
```

Entries such as the following occur on startup or when some event occurs such as an interface transitioning or a user manually clearing the routing table.

```
IGRP: broadcasting request on Ethernet0  
IGRP: broadcasting request on Ethernet1
```

The following type of entry can result when routing updates become corrupted between sending and receiving routers.

```
IGRP: bad checksum from 160.89.64.43
```

An entry such as the following should never appear. If it does, the receiving router has a bug in the software or a problem with the hardware. In either case, contact your technical support representative.

```
IGRP: system 45 from 160.89.64.234, should be system 109
```

Debug IP-IGRP-Events

Use the **debug ip-igrp-events** command to display summary IGRP routing messages that indicate the source and destination of each update, as well as the number of routes in each update. Messages are not generated for each route.

Syntax for the **debug ip-igrp-events** command follows.

debug ip-igrp-events [*ip-address*]

If the IP address of an IGRP neighbor is specified, the resulting **debug ip-igrp-events** output will include messages describing updates from that neighbor and updates that the router broadcasts toward that neighbor.

This command is particularly useful when there are many networks in your routing table. In this case, using **debug ip-igrp** could flood the console and make the router unusable. Use **debug ip-igrp-events** instead to display summary routing information.

Figure 10-20 shows example **debug ip-igrp** output.

```
IGRP: sending update to 255.255.255.255 via Ethernet1 (160.89.33.8)
IGRP: Update contains 26 interior, 40 system, and 3 exterior routes.
IGRP: Total routes in update: 69
IGRP: sending update to 255.255.255.255 via Ethernet0 (160.89.32.8)
IGRP: Update contains 1 interior, 0 system, and 0 exterior routes.
IGRP: Total routes in update: 1
IGRP: received update from 160.89.32.24 on Ethernet0
IGRP: Update contains 17 interior, 1 system, and 0 exterior routes.
IGRP: Total routes in update: 18
IGRP: received update from 160.89.32.7 on Ethernet0
IGRP: Update contains 5 interior, 1 system, and 0 exterior routes.
IGRP: Total routes in update: 6
```

Updates sent to these two destination addresses

Updates received from these source addresses

Figure 10-21 Example Debug IP-IGRP Output

Figure 10-20 shows that the router has sent two updates to the broadcast address 255.255.255.255. The router also received two updates. Three lines of output describe each of these updates. Explanations for representative lines of output from Figure 10-20 follow.

The first line of output, as follows, indicates whether the router sent or received the update packet, the source or destination address, and the interface through which the update was sent or received. If the update was sent, the IP address assigned to this interface is shown (in parentheses).

```
IGRP: sending update to 255.255.255.255 via Ethernet1 (160.89.33.8)
```

The second line of output summarizes the number and types of routes described in the update.

```
IGRP: Update contains 26 interior, 40 system, and 3 exterior routes.
```

The third line of output indicates the total number of routes described in the update.

```
IGRP: Total routes in update: 69
```

Debug IP-OSPF-Events

Use the **debug ip-ospf-events** command to generate information concerning OSPF-related events, such as adjacencies, flooding information, designated router selection, and SPF calculation.

Figure 10-22 shows example **debug ip-ospf-events** output.

```
OSPF:hello with invalid timers on interface Ethernet0
hello interval received 10  configured 10
net mask received 255.255.255.0  configured 255.255.255.0
dead interval received 40  configured 30
```

Figure 10-22 Example Debug IP-OSPF-Events Output

The **debug ip-ospf-events** output shown in Figure 10-22 might appear if any of the following occurs:

- The IP subnet masks for routers on the same network do not match.
- The OSPF Hello interval for the router does not match that configured for a neighbor.
- The OSPF Dead interval for the router does not match that configured for a neighbor.

If a router configured for OSPF routing is not seeing an OSPF neighbor on an attached network, do the following:

- Make sure that both routers have been configured with the same IP mask, OSPF Hello interval, and OSPF dead interval.
- Make sure that the both neighbors are part of same kind of area.

In the following example line, the neighbor and this router are not part of a stub area (that is, one is a part of transit area and the other is a part of stub area, as explained on page 92 of RFC 1247).

```
OSPF: hello packet with mismatched E bit
```

Debug IP-Packet

Use the **debug ip-packet** command to display general IP debugging information and IPSO security transactions.

Note: Because the **debug ip-packets** command generates a fair amount of output, you should only use it when traffic on the IP network is low.

If a communication session is closing when it should not be, an end-to-end connection problem can be the cause. The **debug ip-packet** command is useful for analyzing the messages traveling between the local and remote hosts.

IP debugging information includes packets received, generated, and forwarded. Fast-switched packets do not generate messages.

IPSO security transactions include messages that describe the cause of failure each time a datagram fails a security test in the system. This information also is sent to the sending host when the router configuration allows it.

The syntax for the **debug ip-packet** command follows:

debug ip-packet [*list*]

In this syntax statement, *list* is an optional IP access *list* that you can specify. If the datagram is not permitted by that access list, the related debugging output is suppressed.

Figure 10-23 shows example **debug ip-packet** output.

```
IP: s=131.108.13.44 (Fddi0), d=157.125.254.1 (Serial2), g=131.108.16.2, forward
IP: s=131.108.1.57 (Ethernet4), d=192.36.125.2 (Serial2), g=131.108.16.2, forward
IP: s=131.108.1.6 (Ethernet4), d=255.255.255.255, rcvd 2
IP: s=131.108.1.55 (Ethernet4), d=131.108.2.42 (Fddi0), g=131.108.13.6, forward
IP: s=131.108.89.33 (Ethernet2), d=131.130.2.156 (Serial2), g=131.108.16.2, forward
IP: s=131.108.1.27 (Ethernet4), d=131.108.43.126 (Fddi1), g=131.108.23.5, forward
IP: s=131.108.1.27 (Ethernet4), d=131.108.43.126 (Fddi0), g=131.108.13.6, forward
IP: s=131.108.20.32 (Ethernet2), d=255.255.255.255, rcvd 2
IP: s=131.108.1.57 (Ethernet4), d=192.36.125.2 (Serial2), g=131.108.16.2, access denied
```

Figure 10-23 Example Debug IP-Packet Output

Figure 10-23 shows two types of messages that **debug ip-packet** can produce; the first line of output describes an IP packet that the router forwards, and the third line of output describes a packet that is destined for the router. In the third line of output, “rcvd 2” indicates that the router decided to receive the packet.

Table 10-17 describes the fields shown in the first line of Figure 10-23.

Table 10-17 Debug IP-Packet Field Descriptions

Field	Description
IP:	Indicates that this is an IP packet
s = 131.108.13.44 (Fddi0)	Indicates the source address of the packet and the name of the interface that received the packet
d = 157.125.254.1 (Serial2)	Indicates the destination address of the packet and the name of the interface (in this case, S2)through which the packet is being sent out on the network
g = 131.108.16.2	Indicates the address of the next hop gateway
forward	Indicates that the router is forwarding the packet. If a filter denies a packet, “access denied” replaces “forward,” as shown in the last line of output in Figure 10-23.

The calculation on whether to send an security error message can be somewhat confusing. It depends upon both the security label in the datagram and the label of the incoming interface. First, the label contained in the datagram is examined for anything obviously wrong. If nothing is wrong, it should be assumed to be correct. If there is something wrong, the datagram should be treated as *unclassified genser*. Then this label is compared with the interface range, and the appropriate action is taken, as Table 10-18 describes.

Table 10-18 Security Actions

Classification	Authorities	Action Taken
Too low	Too low	No Response
	Good	No Response
	Too high	No Response
In range	Too low	No Response
	Good	Accept
	Too high	Send Error
Too high	Too Low	No Response
	In range	Send Error
	Too high	Send Error

The range of ICMP error messages that can be generated by the security code is very small. The only possible error messages and their meanings follow:


- “ICMP Parameter problem, code 0”—Error at pointer.
- “ICMP Parameter problem, code 1”—Missing option.
- “ICMP Parameter problem, code 2”—See Note that follows.
- “ICMP Unreachable, code 10”—Administratively prohibited.

Note: The message “ICMP Parameter problem, code 2” identifies a very specific error that occurs in the processing of a datagram. This message indicates that the router received a datagram containing a maximum length IP header, but no security option. After being processed and routed to another interface, it is discovered that the outgoing interface is marked with “add a security label.” Since the IP header is already full, the system cannot add a label and must drop the datagram and return an error message.

Debug IP-RIP

Use the **debug ip-rip** command to enable logging of RIP routing transactions.

Figure 10-24 shows example **debug ip-rip** output.



```
RIP: received update from 160.89.80.28 on Ethernet0
      160.89.95.0 in 1 hops
      160.89.81.0 in 1 hops
      160.89.66.0 in 2 hops
      131.108.0.0 in 16 hops (inaccessible)
      0.0.0.0 in 7 hop
RIP: sending update to 255.255.255.255 via Ethernet0 (160.89.64.31)
      subnet 160.89.94.0, metric 1
      131.108.0.0 in 16 hops (inaccessible)
RIP: sending update to 255.255.255.255 via Serial1 (160.89.94.31)
      subnet 160.89.64.0, metric 1
      subnet 160.89.66.0, metric 3
      131.108.0.0 in 16 hops (inaccessible)
      default 0.0.0.0, metric 8
```

Figure 10-24 Example Debug IP-RIP Output

Figure 10-24 shows that the router being debugged has received updates from one router at source address 160.89.80.28. That router sent information about five destinations in the routing table update. Notice that the fourth destination address in the update—131.108.0.0—is inaccessible because it is more than 15 hops away from the router sending the update. The router being debugged also sent updates—in both cases to the broadcast address 255.255.255.255 as the destination address.

The first line in Figure 10-24 is self-explanatory.

The second line in Figure 10-24 is an example of a routing table update. It shows how many hops a given Internet address is from the router.

The entries in Figure 10-24 showing that the router is sending updates are similar, except that the number in parentheses is the source address encapsulated into the IP header.

Examples of additional output that the **debug ip-rip** command can generate follow.

Entries such as the following appear at startup or when some event occurs such as an interface transitioning or the user manually clearing the routing table.

```
RIP: broadcasting general request on Ethernet0
RIP: broadcasting general request on Ethernet1
```

The following line is self-explanatory.

```
RIP: received request from 160.89.80.207 on Ethernet0
```

An entry such as the following is most likely caused by a malformed packet from the transmitter.

```
RIP: bad version 128 from 160.89.80.43
```

Debug IP-TCP

Use the **debug ip-tcp** command to display significant TCP transactions such as state changes, retransmissions, and duplicate packets.

This command is particularly useful for debugging a performance problem on a TCP/IP network that you have isolated above the data link layer.

The **debug ip-tcp** command displays output for packets the router sends and receives, but does not display output for packets it forwards.

Figure 10-25 shows example **debug ip-tcp** output.

```
TCP: sending SYN, seq 168108, ack 88655553
TCP0: Connection to 26.9.0.13:22530, advertising MSS 966
TCP0: state was LISTEN -> SYNSENT [23 -> 26.9.0.13(22530)]
TCP0: state was SYNSENT -> SYNRCVD [23 -> 26.9.0.13(22530)]
TCP0: Connection to 26.9.0.13:22530, received MSS 956
TCP0: restart retransmission in 5996
TCP0: state was SYNRCVD -> ESTAB [23 -> 26.9.0.13(22530)]
TCP2: restart retransmission in 10689
TCP2: restart retransmission in 10641
TCP2: restart retransmission in 10633
TCP2: restart retransmission in 13384 -> 26.0.0.13(16151)]
TCP0: restart retransmission in 5996 [23 -> 26.0.0.13(16151)]
```

Figure 10-25 Example Debug IP-TCP Output

Table 10-19 describes significant fields shown in Figure 10-25.

Table 10-19 Debug IP-TCP Field Descriptions

Field	Description
TCP:	Indicates that this is a TCP transaction.
sending SYN	Indicates that a synchronize packet is being sent.
seq 168108	Indicates the sequence number of the data being sent.
ack 88655553	Indicates the sequence number of the data being acknowledged.
TCP0:	Indicates the TTY number (0, in this case) with which this TCP connection is associated.
Connection to 26.9.0.13:22530	Indicates the remote address with which a connection has been established.
advertising MSS 966	Indicates the maximum segment size the TCP connection will allow.
state was LISTEN -> SYNSENT	Indicates that the TCP state machine changed state from LISTEN to SYNSENT. Possible TCP states follow. <ul style="list-style-type: none"> ■ ESTAB—Connection established. ■ LISTEN—Listening for a connection request. ■ SYNRCVD—Received a SYN packet. ■ SYNSENT—Sent a SYN packet.
[23 -> 26.9.0.13(22530)]	Within these brackets: <ul style="list-style-type: none"> ■ The first field (23) indicates local TCP port. ■ The second field (26.9.0.13) indicates the destination IP address. ■ The third field (22530) indicates the destination TCP port.
restart retransmission in 5996	Indicates the number of milliseconds until the next retransmission takes place.

Debug LAPB

Use the **debug lapb** command to display all traffic for interfaces using LAPB encapsulation. This command displays information on the X.25 layer 2 protocol. It is useful to users who are familiar with the LAPB protocol.

You can use **debug lapb** to determine why X.25 virtual circuits or LAPB connections are going up and down. It is also useful for identifying link problems, as evidenced when **show interface** displays a high number of rejects or frame errors over the X.25 link.



Caution: Because the **debug lapb** command generates a lot of output, you should only use it when the aggregate of all LAPB traffic on X.25 and LAPB interfaces is less than five frames per second.

Figure 10-26 shows example **debug lapb** output.

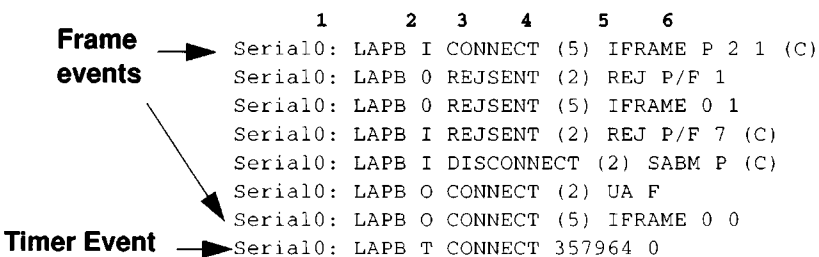


Figure 10-26 Example Debug LAPB Output—Part 1

In Figure 10-26, each line of output describes a LAPB event. There are two types of LAPB events: frame events (when a frame enters or exits the router) and timer events. In Figure 10-26, the last line describes a timer event; all of the other lines describe frame events. Table 10-20 describes the first six fields shown in Figure 10-26.

Table 10-20 Debug LAPB Field Descriptions—Part 1

Field	Description
First field	Interface type and unit number reporting the frame event.
Second field	Protocol providing the information.
Third field	Type of frame event. Possible values follow: <ul style="list-style-type: none"> ■ I—Frame input ■ O—Frame output ■ T—T1 timer expired
Fourth field	State of the protocol when the frame event occurred. Possible values follow: <ul style="list-style-type: none"> ■ BUSY ■ CONNECT ■ DISCONNECT ■ DISCSENT (disconnect sent) ■ ERROR (FRMR frame sent) ■ REJSENT (reject frame sent) ■ SABMSENT (SABM frame sent)
Fifth field	In a frame event, this value is the size of the frame (in bytes). In a timer event, this value is the current timer value (in milliseconds).

Field	Description
Sixth field	<p>In a frame event, this value is the frame type name. Possible values for frame type names follow:</p> <ul style="list-style-type: none"> ■ DISC—Disconnect ■ DM—Disconnect mode ■ FRMR—Frame reject ■ IFRAME—Information frame ■ ILLEGAL—Illegal LAPB frame ■ REJ—Reject ■ RNR—Receiver not ready ■ RR—Receiver ready ■ SABM—Set asynchronous balanced mode ■ UA—Unnumbered acknowledgment <p>In a timer event, this value is the number of retransmissions already attempted.</p>
Subsequent fields	<p>As Figure 10-26 shows, a timer event only displays the first six fields of debug lapb output. For frame events, however, the fields that follow the sixth field document the LAPB control information present in the frame. Depending on the value of the frame type name shown in the sixth field, these fields may or may not appear. Descriptions of the fields following the first six fields shown in Figure 10-26 follow.</p>

If the frame's Poll/Final bit is set, an indicator will be printed after the frame type name. Possible values follow:

- F—Final (printed for Response frames)
- P—Poll (printed for Command frames)
- P/F—Poll/final (printed for RR, RNR and REJ frames, which can be either Command or Response frames)

After the Poll/Final indicator, depending on the frame type, three different types of LAPB control information can be printed.

For information frames, the value of the N(S) field and the N(R) field will be printed. The N(S) field of an information frame is the sequence number of that frame, so this field will rotate between 0 and 7 for successive outgoing information frames and (under normal circumstances) also will rotate for incoming information frame streams. The N(R) field is a "piggybacked" acknowledgement for the incoming information frame stream; it informs the other end of the link what sequence number is expected next.

RR, RNR and REJ frames have an N(R) field, so the value of that field is printed. This field has exactly the same significance that it does in an information frame.

For the FRMR frame, the frame's three bytes of error information is printed (in hexadecimal).

The remaining frames do not have these data, so nothing will be printed.

Finally, for incoming frames, the last field will indicate if the received frame was a command (C) or a response (R).

Debug LNM-Events

Use the **debug lnm-events** command to display any unusual events that occur on a Token Ring network. This includes such things as stations reporting errors, error thresholds being exceeded, and so on.

Figure 10-27 shows sample **debug lnm-events** output.

```
IBMNM3: Adding 0000.3001.1166 to error list
IBMNM3: Station 0000.3001.1166 going into preweight condition
IBMNM3: Station 0000.3001.1166 going into weight condition
IBMNM3: Removing 0000.3001.1166 from error list
LANMGR0: Beaconsing is present on the ring.
LANMGR0: Ring is no longer beaconsing
IBMNM3: Beaconsing, Postmortem Started
IBMNM3: Beaconsing, heard from 0000.3000.1234
IBMNM3: Beaconsing, Postmortem Next Stage
IBMNM3: Beaconsing, Postmortem Finished
```

Figure 10-27 Example Debug LMN-Events Output

Explanations for the messages shown in Figure 10-27 follow.

The following message indicates that station 0000.0300.1234 reported errors and has been added to the list of stations reporting errors. This station is located on Ring 3.

```
IBMNM3: Adding 0000.0300.1234 to error list
```

The following message indicates that the station has passed the “early warning” threshold for error counts.

```
IBMNM3: Station 0000.0300.1234 going into preweight condition
```

The following message indicates that station 0000.0300.1234 is experiencing a severe number of errors.

```
IBMNM3: Station 0000.0300.1234 going into weight condition
```

The following message indicates that the station’s error counts have all decayed to zero, so this station is being removed from the list of stations that have reported errors.

```
IBMNM3: Removing 0000.0300.1234 from error list
```

The following message indicates that Ring 0 has entered failure mode. This ring number is assigned internally.

```
LANMGR0: Beaconsing is present on the ring.
```

The following message indicates that Ring 0 is no longer in failure mode. This ring number is assigned internally.

```
LANMGR0: Ring is no longer beaconsing
```

The following message indicates that the router is beginning its attempt to determine whether or not any stations left the ring during the automatic recovery process for the last beaconing failure. The router attempts to contact stations that were part of the fault domain to see if they are still operating on the ring.

```
IBMNM3: Beaconing, Postmortem Started
```

The following message indicates that the router is attempting to determine whether or not any stations left the ring during the automatic recovery process for the last beaconing failure. It heard back from station 0000.0300.1234, one of the two stations in the fault domain.

```
IBMNM3: Beaconing, heard from 0000.0300.1234
```

The following message indicates that the router is attempting to determine whether or not any stations left the ring during the automatic recovery process for the last beaconing failure. It is initiating another attempt to contact the two stations in the fault domain.

```
IBMNM3: Beaconing, Postmortem Next Stage
```

The following output indicates that the router has attempted to determine whether or not any stations left the ring during the automatic recovery process for the last beaconing failure. It has successfully heard back from both stations that were part of the fault domain.

```
IBMNM3: Beaconing, Postmortem Finished
```

Explanations for other messages that the **debug lnm-events** command can generate follow.

The following message indicates that the router is out of memory.

```
LANMGR: memory request failed, find_or_build_station()
```

The following message indicates that Ring number 3 is experiencing a large number of errors that cannot be attributed to any individual station.

```
IBMNM3: Non-isolating error threshold exceeded
```

The following message indicates that a station (or stations) on Ring 3 are receiving frames faster than they can be processed.

```
IBMNM3: Adapters experiencing congestion
```

The following message indicates that the beaconing has lasted for over one minute and is considered to be a “permanent” error.

```
IBMNM3: Beaconing, permanent
```

The following message indicates that the beaconing lasted for less than one minute. The router is attempting to determine whether either of the stations in the fault domain left the ring.

```
IBMNM: Beaconing, Destination Started
```

In the preceding line of output, the following can replace Started: Next State; Finished; Timed out; and cannot find station 0000.0301.4876.

Debug LNM-LLC

Use the **debug lnm-llc** command to display all communication between the router/bridge and the LNM that have connections to it. One line is displayed for each message sent or received.

Figure 10-28 shows sample **debug lnm-llc** output.

```
IBMNM: Received LRM Set Reporting Point frame from 1000.5ade.0d8a.
IBMNM: found bridge: 001-2-00A, addresses: 0000.3040.a630 4000.3040.a630
IBMNM: Opening connection to 1000.5ade.0d8a on TokenRing0
IBMNM: Sending LRM LAN Manager Accepted to 1000.5ade.0d8a on link 0.
IBMNM: sending LRM New Reporting Link Established to 1000.5a79.dbf8 on link 1.
IBMNM: Determining new controlling LNM
IBMNM: Sending Report LAN Manager Control Shift to 1000.5ade.0d8a on link 0.
IBMNM: Sending Report LAN Manager Control Shift to 1000.5a79.dbf8 on link 1.

IBMNM: Bridge 001-2-00A received Request Bridge Status from 1000.5ade.0d8a.
IBMNM: Sending Report Bridge Status to 1000.5ade.0d8a on link 0.
IBMNM: Bridge 001-2-00A received Request REM Status from 1000.5ade.0d8a.
IBMNM: Sending Report REM Status to 1000.5ade.0d8a on link 0.
IBMNM: Bridge 001-2-00A received Set Bridge Parameters from 1000.5ade.0d8a.
IBMNM: Sending Bridge Parameters Set to 1000.5ade.0d8a on link 0.
IBMNM: sending Bridge Params Changed Notification to 1000.5a79.dbf8 on link 1.
IBMNM: Bridge 001-2-00A received Set REM Parameters from 1000.5ade.0d8a.
IBMNM: Sending REM Parameters Set to 1000.5ade.0d8a on link 0.
IBMNM: sending REM Parameters Changed Notification to 1000.5a79.dbf8 on link 1.
IBMNM: Bridge 001-2-00A received Set REM Parameters from 1000.5ade.0d8a.
IBMNM: Sending REM Parameters Set to 1000.5ade.0d8a on link 0.
IBMNM: sending REM Parameters Changed Notification to 1000.5a79.dbf8 on link 1.
IBMNM: Received LRM Set Reporting Point frame from 1000.5ade.0d8a.
IBMNM: found bridge: 001-1-00A, addresses: 0000.3080.2d79 4000.3080.2d79
```

Figure 10-28 Example Debug LMN-LLC Output

As Figure 10-28 indicates, **debug lnm-llc** output can vary somewhat in format. Table 10-21 describes significant fields shown in the first line of output in Figure 10-28.

Table 10-21 Debug LNM-LLC Field Descriptions

Field	Description
IBMNM:	Indicates that this line of output displays LLC-level debugging information.
Received	Indicates that the router received a frame. The other possible value is Sending, to indicate that the router is sending a frame.
RS	Indicates which function of the LLC-level software is communicating: <ul style="list-style-type: none"> ■ CRS—Configuration Report Server ■ LBS—LAN Bridge Server ■ LRM—LAN Reporting Manager ■ REM—Ring Error Monitor ■ RPS—Ring Parameter Server ■ RS—Ring Station
Set Reporting Point	Name of the specific frame that the router sent or received. Possible values include the following: <ul style="list-style-type: none"> ■ Bridge Counter Report ■ Bridge Parameters Set ■ Bridge Parameters Changed Notification ■ CRS Remove Ring Station ■ CRS Request Station Information ■ CRS Report Station Information ■ CRS Report NAUN Change ■ CRS Ring Station Removed ■ LRM LAN Manager Accepted ■ New Reporting Link Established ■ REM Forward MAC Frame ■ REM Parameters Changed Notification ■ REM Parameters Set ■ Report Bridge Status ■ Report LAN Manager Control Shift ■ Report REM Status ■ Request Bridge Status

Field	Description
Set Reporting Point (continued)	<ul style="list-style-type: none"> ■ Request REM Status ■ Set Bridge Parameters ■ Set REM Parameters ■ LRM Set Reporting Point
from 4000.3040.a670	If the router has received the frame, this address is the source address of the frame. If the router is sending the frame, this address is the destination address of the frame.

Explanations for other types of messages shown in Figure 10-28 follow.

The following message indicates that the lookup for the bridge with which the LAN Manager was requesting to communicate was successful.

```
IBMNM: found bridge: 001-2-00A, addresses: 0000.3040.a630 4000.3040.a630
```

The following message is self-explanatory.

```
IBMNM: Opening connection to 1000.5ade.0d8a on TokenRing0
```

The following message indicates that a LAN Manager has connected or disconnected from an internal bridge, and that the router computes which LAN Manager is allowed to change parameters.

```
IBMNM: Determining new controlling LNM
```

The following line of output indicates which bridge in the router is the destination for the frame.

```
IBMNM: Bridge 001-2-00A received Request Bridge Status from 1000.5ade.0d8a.
```

Debug LNM-MAC

Use the **debug lnm-mac** command to display all management communication between the router/bridge and all stations on the local Token Rings. One line is displayed for each message sent or received.

Figure 10-29 shows sample **debug lnm-mac** output.

```
LANMGR0: RS received request address from 4000.3040.a670.  
LANMGR0: RS sending report address to 4000.3040.a670.  
LANMGR0: RS received request state from 4000.3040.a670.  
LANMGR0: RS sending report state to 4000.3040.a670.  
LANMGR0: RS received request attachments from 4000.3040.a670.  
LANMGR0: RS sending report attachments to 4000.3040.a670.  
LANMGR2: RS received ring purge from 0000.3040.a630.  
LANMGR2: CRS received report NAUN change from 0000.3040.a630.  
LANMGR2: RS start watching ring poll.  
LANMGR0: CRS received report NAUN change from 0000.3040.a630.  
LANMGR0: RS start watching ring poll.  
LANMGR2: REM received report soft error from 0000.3040.a630.  
LANMGR0: REM received report soft error from 0000.3040.a630.  
LANMGR2: RS received ring purge from 0000.3040.a630.  
LANMGR2: RS received AMP from 0000.3040.a630.  
LANMGR2: RS received SMP from 0000.3080.2d79.  
LANMGR2: CRS received report NAUN change from 1000.5ade.0d8a.  
LANMGR2: RS start watching ring poll.  
LANMGR0: RS received ring purge from 0000.3040.a630.  
LANMGR0: RS received AMP from 0000.3040.a630.  
LANMGR0: RS received SMP from 0000.3080.2d79.  
LANMGR0: CRS received report NAUN change from 1000.5ade.0d8a.  
LANMGR0: RS start watching ring poll.  
LANMGR2: RS received SMP from 1000.5ade.0d8a.  
LANMGR2: RPS received request initialization from 1000.5ade.0d8a.  
LANMGR2: RPS sending initialize station to 1000.5ade.0d8a.
```

Figure 10-29 Example Debug LNM-MAC Output

Table 10-21 describes significant fields shown in the first line of output in Figure 10-29.

Table 10-22 Debug LNM-MAC Field Descriptions

Field	Description
LANMGR0:	LANMGR indicates that this line of output displays MAC-level debugging information. 0 indicates the number of the Token Ring interface associated with this line of debugging output.
RS	Indicates which function of the MAC level software is communicating: <ul style="list-style-type: none"> ■ CRS—Configuration Report Server ■ REM—Ring Error Monitor ■ RPS—Ring Parameter Server ■ RS—Ring Station
received	Indicates that the router received a frame. The other possible value is sending, to indicate that the router is sending a frame.
request address	Name of the specific frame that the router sent or received. Possible values include the following: <ul style="list-style-type: none"> ■ AMP ■ Initialize station ■ Report address ■ Report attachments ■ Report NAUN change ■ Report soft error ■ Report state ■ Request address ■ Request attachments ■ Request initialization ■ Request state ■ Ring purge ■ SMP
from 4000.3040.a670	If the router has received the frame, this address is the source address of the frame. If the router is sending the frame, this address is the destination address of the frame.

As Figure 10-29 indicates, all **debug lnm-mac** messages follow the format described in Table 10-21 except the following:

```
LANMGR2: RS start watching ring poll
LANMGR2: RS stop watching ring poll
```

This message indicates that the router stops receiving AMP and SMP frames. These frames are used to build a current picture of which stations are on the ring.

Debug Local-ACK-State

Use the **debug local-ack-state** command to print out the new and the old state conditions whenever there is a state change in the Local Acknowledgment state machine.

Figure 10-30 shows example **debug local-ack-state** output.

```
LACK_STATE: 2370300, hashp 2AE628, old state = disconn, new state = awaiting LLC2 open to finish
LACK_STATE: 2370304, hashp 2AE628, old state = awaiting LLC2 open to finish, new state = connected
LACK_STATE: 2373816, hashp 2AE628, old state = connected, new state = disconnected
LACK_STATE: 2489548, hashp 2AE628, old state = disconn, new state = awaiting LLC2 open to finish
LACK_STATE: 2489548, hashp 2AE628, old state = awaiting LLC2 open to finish, new state = connected
LACK_STATE: 2490132, hashp 2AE628, old state = connected, new state = awaiting linkdown response
LACK_STATE: 2490140, hashp 2AE628, old state = awaiting linkdown response, new state = disconnected
LACK_STATE: 2497640, hashp 2AE628, old state = disconn, new state = awaiting LLC2 open to finish
LACK_STATE: 2497644, hashp 2AE628, old state = awaiting LLC2 open to finish, new state = connected
```

Figure 10-30 Example Debug Local-ACK-State Output

Table 10-23 describes significant fields shown in Figure 10-30.

Table 10-23 Debug Local-ACK-State Field Descriptions

Field	Description
LACK_STATE:	Indicates that this packet describes a state change in the Local Acknowledgment state machine.
2370300	System clock
hashp 2AE628	Internal control block pointer used by technical support staff for debugging purposes.
old state = disconnected	Indicates the old state condition in the Local Acknowledgment state machine. Possible values include: <ul style="list-style-type: none">■ Disconnected■ Awaiting LLC2 open to finish■ Connected■ Awaiting linkdown response■ Disconnected
new state = awaiting LLC2 open to finish	Indicates the new state condition in the Local Acknowledgment state machine. Possible values include: <ul style="list-style-type: none">■ Disconnected■ Awaiting LLC2 open to finish■ Connected■ Awaiting linkdown response■ Disconnected

Debug Novell-Packet

Use the **debug novell-packet** command to output information about packets received, transmitted, and forwarded.

This command is useful for learning whether Novell packets are traveling over a router.

Note: In order to generate **debug novell-packet** information on all Novell traffic traveling over the router, you must first configure the router so that fast switching is disabled. Use the **no novell route-cache** command on all interfaces on which you want to observe traffic. If the router is configured for Novell fast-switching, only Novell broadcast packets (SAP, RIP, and Novell NetBIOS) will be displayed.

Figure 10-31 shows example **debug novell-packet** output.

```
Novell: src=160.0260.8c4c.4f22, dst=1.0000.0000.0001, packet received
Novell: src=160.0260.8c4c.4f22, dst=1.0000.0000.0001,gw=183.0000.0c01.5d85,
sending packet
```

Figure 10-31 Example Debug Novell-Packet Output

In Figure 10-31, the first line indicates that the router receives a packet from a Novell station (address 160.0260.8c4c.4f22); this trace does not indicate the address of the immediate router sending the packet to this router. In the second line, the router forwards the packet toward the Novell server (address 1.0000.0000.0001) through an immediate router (183.0000.0c01.5d85).

Table 10-24 describes significant fields shown in Figure 10-31.

Table 10-24 Debug Novell-Packet Field Descriptions

Field	Description
Novell	Shows that this is a Novell IPX packet.
src = 160.0260.8c4c.4f22	Source address of the Novell packet. The Novell network number is 160. Its MAC address is 0260.8c4c.4f22.
dst = 1.0000.0000.0001	Destination address for the Novell packet. The address 0000.0000.0001 is an internal MAC address, and the network number 1 is the internal network number of a Novell 3.11 server.
packet received	The router received this packet from a Novell station, possibly through an intermediate router.
gw = 183.0000.0c01.5d85	The router is sending the packet over to the next hop router; its address of 183.0000.0c01.5d85 was learned from the Novell routing table.
sending packet	The router is attempting to send this packet.

Debug Novell-Routing

Use the **debug novell-routing** command to print out information on Novell routing packets that the router sends and receives.

Figure 10-32 shows example **debug novell-routing** output.

```
NovellRIP: update from 9999.0260.8c6a.1733
          110801 in 1 hops, delay 2
NovellRIP: sending update to 12FF02:ffff.ffff.ffff via Ethernet
          network 555, metric 2, delay 3
          network 1234, metric 3, delay 4
```

Figure 10-32 Example Debug Novell-Routing Output

Table 10-25 describes significant fields shown in Figure 10-32.

Table 10-25 Debug Novell-Routing Field Descriptions

Field	Description
NovellRIP	Shows that this is a Novell RIP packet.
update from 9999.0260.8c6a.1733	Indicates that this packet is a routing update from a Novell server at address 9999.0260.8c6a.1733.
110801 in 1 hops	Indicates that network 110801 is one hop away from the router at address 9999.0260.8c6a.1733.
delay 2	A time measurement (1/18th second) that the NetWare shell uses to estimate how long to wait for a response from a file server. Also known as ticks.
sending update to 12FF02:ffff.ffff.ffff via Ethernet 1	The router is sending this Novell routing update packet to address 12FF02:ffff.ffff.ffff through its Ethernet 1 interface.
network 555	Indicates that the packet includes routing update information for network 555.
metric 2	Indicates that network 555 is two metrics (or hops) away from the router.
delay 3	Indicates that network 555 is a delay of 3 away from the router. Delay is a measurement that the NetWare shell uses to estimate how long to wait for a response from a file server. Also known as ticks.

Debug Novell-SAP

Use the **debug novell-sap** command to display additional information about Novell Service Advertisement (SAP) packets.

Normally, a router or server sends out one SAP update per minute. Each SAP packet can include up to seven entries. If many servers are advertising on the network, the router sends out multiple packets per update. For example, if a router has 20 entries in the SAP table, it would send three SAP packets per update. The first SAP would include the first seven entries, the second SAP would include the next seven entries, and the last update would include the last six entries.

Figure 10-33 shows example **debug novell-sap** output.

```
NovellSAP: at 0023F778:  
I SAP Response type 0x2 len 160 src:160.0000.0c00.070d dest:160.ffff.ffff.ffff(452)  
  type 0x4, "HELLO2", 199.0002.0004.0006 (451), 2 hops  
  type 0x4, "HELLO1", 199.0002.0004.0006 (451), 2 hops  
NovellSAP: sending update to 160  
NovellSAP: at 00169080:  
  O SAP Update type 0x2 len 96 ssoc:0x452 dest:160.ffff.ffff.ffff(452)  
Novell: type 0x4, "Magnolia", 42.0000.0000.0000 (451), 2 hops
```

←
**Describes
a single
SAP
packet**

Figure 10-33 Example Debug Novell-SAP Output

As Figure 10-33 shows, the **debug novell-sap** command generates multiple lines of output for each SAP packet—a packet summary message and a service detail message.

Explanations for representative lines of output from Figure 10-33 follow.

The first line of output displays the internal router memory address of the packet. Cisco support staff use this information in problem debugging.

```
NovellSAP: at 0023F778:
```

Table 10-26 describes the fields shown in the second line of output in Figure 10-33.

Table 10-26 Debug Novell-SAP Field Descriptions—Part 1

Field	Description
I	Indicates whether the router received the SAP packet as input (I) or is sending an update (O).
SAP Response type 0x2	Indicates the packet type. Format is 0xn; possible values for <i>n</i> include: <ul style="list-style-type: none"> ■ 1—General query ■ 2—General response ■ 3—Get nearest server request ■ 4—Get nearest server response
len 160	Length of this packet (in bytes).
src: 160.000.0c00.070d	Indicates the source address of the packet.
dest: 160.ffff.ffff.ffff	Indicates the Novell network number and broadcast address of the destination Novell network for which the message is intended.
(452)	Novell socket number of the process sending the packet at the source address. This number is always 452, which is the socket number for the SAP process.

Table 10-26 describes the fields shown in the third line of output in Figure 10-33.

Table 10-27 Debug Novell-SAP Field Descriptions—Part 2

Field	Description
type 0x4	<p>Indicates the type of service the server sending the packet provides. Format is 0x<i>n</i>. Some of the values for <i>n</i> are proprietary to Novell. Those values for <i>n</i> that have been published include:</p> <ul style="list-style-type: none"> ■ 0—Unknown ■ 1—User ■ 2—User group ■ 3—Print queue ■ 4—File server ■ 5—Job server ■ 6—Gateway ■ 7—Print server ■ 8—Archive queue ■ 9—Archive server ■ A—Job queue ■ B—Administration ■ 24—Remote bridge server ■ 47—Advertising print server <p>Contact Novell for more information.</p>
“HELLO2”	Name of the server being advertised.
199.0002.0004.0006 (451)	Indicates the network number and address (and socket) of the server generating the SAP packet.
2 hops	Number of hops to the server from the router.

The fifth line of output, which follows, indicates that the router sent a SAP update to network 160.

```
NovellSAP: sending update to 160
```

As Figure 10-33 shows, the format for **debug novell-sap** output describing a SAP update the router sends is similar to that describing a SAP update the router receives, except that the `ssoc:` field replaces the `src:` field, as the following line of output from Figure 10-33 indicates.

```
O SAP Update type 0x2 len 96 ssoc:0x452 dest:160.ffff.ffff.ffff(452)
```

Table 10-26 describes possible values for the `ssoc:` field.

Table 10-28 Debug Novell-SAP Field Descriptions—Part 3

Field	Description
ssoc:0x452	Indicates the Novell socket number of the process sending the packet at the source address. Possible values include: <ul style="list-style-type: none">■ 451—Network Core Protocol■ 452—Service Advertising Protocol■ 453—Routing Information Protocol■ 455—NetBIOS■ 456—Diagnostics■ 4000 to 6000—Ephemeral sockets used for interaction with file servers and other network communications

Debug Packet

Use the **debug packet** command to log packets that the network is unable to classify.

Figure 10-34 shows example **debug packet** output. Notice how similar it is to **debug broadcast** output.

```
Ethernet0: Unknown ARPA, src 0000.0c00.6fa4, dst ffff.ffff.ffff, type 0x0a0
data 00000c00f23a00000c00ab45, len 60
Serial3: Unknown HDLC, size 64, type 0xaaaa, flags 0x0F00
Serial2: Unknown PPP, size 128
Serial7: Unknown FRAME-RELAY, size 174, type 0x5865, DLCI 7a
Serial0: compressed TCP/IP packet dropped
```

Figure 10-34 Example Debug Packet Output

Table 10-29 describes significant fields shown in Figure 10-34.

Table 10-29 Debug Packet Field Descriptions

Field	Description																				
Ethernet0	Name of the Ethernet interface that received the packet.																				
Unknown	States that the network was unable to classify this packet. Examples include packets with unknown link types.																				
ARPA	States that this packet uses ARPA-style encapsulation. Possible encapsulation styles vary depending on the media, as follows.																				
	<table><tr><th>Media Type</th><th>Encapsulation Style</th></tr><tr><td rowspan="8">Ethernet</td><td>APOLLO</td></tr><tr><td>ARP</td></tr><tr><td>ETHERTALK</td></tr><tr><td>ISO1</td></tr><tr><td>ISO3</td></tr><tr><td>LLC2</td></tr><tr><td>NOVELL-ETHER</td></tr><tr><td>SNAP</td></tr><tr><td rowspan="5">FDDI</td><td>APOLLO</td></tr><tr><td>ISO1</td></tr><tr><td>ISO3</td></tr><tr><td>LLC2</td></tr><tr><td>SNAP</td></tr><tr><td rowspan="2">Frame Relay</td><td>BRIDGE</td></tr><tr><td>FRAME-RELAY</td></tr></table>	Media Type	Encapsulation Style	Ethernet	APOLLO	ARP	ETHERTALK	ISO1	ISO3	LLC2	NOVELL-ETHER	SNAP	FDDI	APOLLO	ISO1	ISO3	LLC2	SNAP	Frame Relay	BRIDGE	FRAME-RELAY
Media Type	Encapsulation Style																				
Ethernet	APOLLO																				
	ARP																				
	ETHERTALK																				
	ISO1																				
	ISO3																				
	LLC2																				
	NOVELL-ETHER																				
	SNAP																				
FDDI	APOLLO																				
	ISO1																				
	ISO3																				
	LLC2																				
	SNAP																				
Frame Relay	BRIDGE																				
	FRAME-RELAY																				

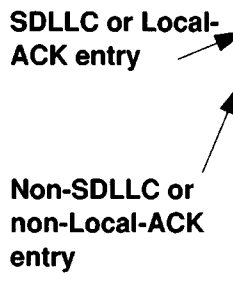
Field	Description	
ARPA (continued)	Serial	BFEX25 BRIDGE DDN-X25 DDNX25-DCE ETHERTALK FRAME-RELAY HDLC HDH LAPB LAPBDCE MULTI-LAPB PPP SDLC-PRIMARY SDLC-SECONDARY SLIP SMDS STUN X25 X25-DCE
	Token Ring	3COM-TR ISO1 ISO3 MAC LLC2 NOVELL-TR SNAP VINES-TR
	Ultramet	ULTRANET ULTRANET-HELLO
src 0000.0c00.6fa4	MAC address of the node generating the packet.	
dst.ffff.ffff.ffff	MAC address of the destination node for the packet.	
type 0x0a0	Packet type.	
data ...	First 12 bytes of the datagram following the MAC header.	
len 60	Length of the message in bytes that the interface received from the wire.	
size 64	Length of the message in bytes that the interface received from the wire. Equivalent to the len field.	
flags 0x0F00	HDLC or PP flags field.	
DLCI 7a	The DLCI number on Frame Relay.	
compressed TCP/IP packet dropped	This message can occur when TCP header compression (THC) is enabled on an interface and the packet does not turn out to be HDLC or X25 after classification.	

Debug RIF

Use the **debug rif** command to provide informational displays for entries entering and leaving the RIF cache.

Note: In order to use the **debug rif** command to display traffic source-routed through an interface, fast switching of SRB frames must first be disabled with the **no source-bridge route-cache** interface subcommand.

Figure 10-35 shows example **debug rif** output.



SDLLC or Local-ACK entry → RIF: U chk da=9000.5a59.04f9,sa=0110.2222.33c1 [4880.3201.00A1.0050] type 8\ on static/remote/0

Non-SDLLC or non-Local-ACK entry → RIF: U chk da=0000.3080.4aed,sa=0000.0000.0000 [] type 8 on TokenRing0/0

RIF: U add 1000.5a59.04f9 [4880.3201.00A1.0050] type 8

RIF: L checking da=0000.3080.4aed, sa=0000.0000.0000

RIF: rcvd TEST response from 9000.5a59.04f9

RIF: U upd da=1000.5a59.04f9,sa=0110.2222.33c1 [4880.3201.00A1.0050]

RIF: rcvd XID response from 9000.5a59.04f9

SR1: sent XID response to 9000.5a59.04f9

Figure 10-35 Example Debug RIF Output

Explanations for representative lines of **debug rif** output in Figure 10-35 follow.

The first line of output in Figure 10-35 is an example of a RIF entry for an interface configured for SDLLC or Local-ACK.

Table 10-30 describes significant fields shown in this line of **debug rif** output.

Table 10-30 Debug RIF Field Descriptions—Part 1

Field	Description
RIF:	Indicates that this message describes RIF debugging output.
U chk	Update checking. The entry is being updated; the timer is set to zero (0).
da = 9000.5a59.04f9	Destination MAC address.
sa = 0110.2222.33c1	Source MAC address. This field contains values of zero (0000.0000.0000) in a non-SDLLC or non-Local-ACK entry.
[4880.3201.00A1.0050]	RIF string. This field is blank (null RIF) in a non-SDLLC or non-Local-ACK entry.
type 8	Possible values follow: <ul style="list-style-type: none"> ■ 0—Null entry ■ 1—This entry was learned from a particular Token Ring port (interface) ■ 2—Statically configured ■ 4—Statically configured for a remote interface ■ 8—This entry is to be aged ■ 16—This entry (which has been learned from a remote interface) is to be aged ■ 32—This entry is not to be aged ■ 64 —This interface is to be used by LAN Network Manager (and is not to be aged)
on static/remote/0	Indicates that this route was learned from a real Token Ring port, in contrast to a virtual ring.

The second line of output in Figure 10-35 is an example of a RIF entry for an interface that is not configured for SDLLC or Local-ACK.

```
RIF: U chk da=0000.3080.4aed,sa=0000.0000.0000 [] type 8 on TokenRing0/0
```

Notice that the source address contains only zero values (0000.0000.0000), and that the RIF string is null ([]). The last element in the entry indicates that this route was learned from a virtual ring, rather than a real Token Ring port.

The third line of output in Figure 10-35 shows that a new entry has been added to the RIF cache.

```
RIF: U add 1000.5a59.04f9 [4880.3201.00A1.0050] type 8
```

The fourth line of output in Figure 10-35 shows that a RIF cache lookup operation has taken place.

```
RIF: L checking da=0000.3080.4aed, sa=0000.0000.0000
```

The fifth line of output in Figure 10-35 shows that a TEST response from address 9000.5a59.04f9 was inserted into the RIF cache.

```
RIF: rcvd TEST response from 9000.5a59.04f9
```

The sixth line of output in Figure 10-35 shows that the RIF entry for this route has been found and updated.

```
RIF: U upd da=1000.5a59.04f9,sa=0110.2222.33c1 [4880.3201.00A1.0050]
```

The seventh line of output in Figure 10-35 shows that an XID response from this address was inserted into the RIF cache.

```
RIF: rcvd XID response from 9000.5a59.04f9
```

The eighth line of output in Figure 10-35 shows that the router sent an XID response to this address.

```
SR1: sent XID response to 9000.5a59.04f9
```

Table 10-31 explains the other possible lines of **debug rif** output.

Table 10-31 Debug RIF Field Descriptions—Part 2

Field	Description
RIF: L Sending XID for <i>address</i>	The router/bridge wanted to send a packet to <i>address</i> but did not find it in the RIF cache. It sent an XID explorer packet to determine which RIF it should use. The attempted packet is dropped.
RIF: L No buffer for XID to <i>address</i>	Similar to the previous display; however, a buffer in which to build the XID packet could not be obtained.
RIF: U remote rif too small [<i>rif</i>]	A packet's RIF was too short to be valid.
RIF: U rej <i>address</i> too big [<i>rif</i>]	A packet's RIF exceeded the maximum size allowed and was rejected. The maximum size is 18 bytes.
RIF: U upd interface <i>address</i>	The RIF entry for this router/bridge's interface has been updated.
RIF: U ign <i>address</i> interface update	A RIF entry that would have updated an interface corresponding to one of this routers interfaces.
RIF: U add <i>address</i> [<i>rif</i>]	The RIF entry for <i>address</i> has been added to the RIF cache.

Field	Description
RIF: U no memory to add rif for <i>address</i>	No memory to add a RIF entry for <i>address</i> .
RIF: removing rif entry for <i>address</i> , <i>type code</i>	The RIF entry for <i>address</i> has been forcibly removed.
RIF: flushed <i>address</i>	The RIF entry for <i>address</i> has been removed because of a RIF cache flush.
RIF: expired <i>address</i>	The RIF entry for <i>address</i> has been aged out of the RIF cache.

Debug Serial-Interface

Use the **debug serial-interface** command to debug a serial connection failure.

If the **show interface** command shows that the line and protocol are down, you can use the **debug serial-interface** command to isolate a timing problem as the cause of a connection failure. If the keepalive values in the mineseq, yourseen, and myseen fields are not incrementing in each subsequent line of output, there is a timing or line problem at one of the ends of the connection.

Note: While the **debug serial-interface** command typically does not generate a lot of output, you nevertheless should use it cautiously during production hours. When SMDS is enabled, for example, it can generate considerable output.

The output of **debug serial-interface** command can vary, depending on the type of WAN configured for an interface: DDR, Frame Relay, HDLC, HSSI, SMDS, or X.25. The output also can vary depending on the type of encapsulation configured for that interface. The hardware platform also can impact **debug serial-interface** output.

The following sections show example **debug serial-interface** displays for various configurations and describe the possible output the command can generate for these configurations.

Debug Serial-Interface for DDR

Table 10-32 describes the error messages the **debug serial-interface** command can generate for a serial interface being used as a V.25bis dialer for dial-on-demand routing.

Table 10-32 Debug Serial-Interface Field Descriptions for DDR

Field	Description
Serial 0: Dialer result = xxxxxxxxxx	This message displays the result returned from the V.25bis dialer. It is useful in debugging if calls are failing. On some hardware platforms, this message cannot be displayed due to hardware limitations. Possible values for the xxxxxxxx variable depend on the V.25bis device with which the router is communicating.
Serial 0: No dialer string defined. Dialing cannot occur.	This message is displayed when a packet is received that should cause a call to be placed. However, there is no dialer string configured, so dialing cannot occur. This message usually indicates a configuration problem.
Serial 0: Attempting to dial xxxxxxxxxx	This message indicates that a packet has been received that passes the dial-on-demand access lists. That packet causes dialing of a phone number. The xxxxxxxx variable is the number being called.
Serial 0: Unable to dial xxxxxxxxxx	This message is displayed if for some reason, the phone call could not be placed. This might be due to a lack of memory, full output queues, or other problems.
Serial 0: disconnecting call	This message is displayed when the router attempts to hang up a call.
Serial 0: idle timeout Serial 0: re-enable timeout Serial 0: wait for carrier timeout	One of these three messages is displayed when their corresponding dialer timer expires. They are mostly informational, but are useful when debugging a disconnected call or call failure.

Debug Serial-Interface for Frame Relay Encapsulation

The following message is displayed if the encapsulation for the interface is frame relay (or HDLC) and the router attempts to send a packet containing an unknown packet type.

```
Illegal serial link type code xxx
```

Debug Serial-Interface for HDLC

Figure 10-36 shows example **debug serial-interface** output for an HDLC connection when keepalives have been enabled.

```
Serial1: HDLC myseq 636119, mineseen 636119, yourseen 515032, line up
Serial1: HDLC myseq 636120, mineseen 636120, yourseen 515033, line up
Serial1: HDLC myseq 636121, mineseen 636121, yourseen 515034, line up
Serial1: HDLC myseq 636122, mineseen 636122, yourseen 515035, line up
Serial1: HDLC myseq 636123, mineseen 636123, yourseen 515036, line up
Serial1: HDLC myseq 636124, mineseen 636124, yourseen 515037, line up
Serial1: HDLC myseq 636125, mineseen 636125, yourseen 515038, line up
Serial1: HDLC myseq 636126, mineseen 636126, yourseen 515039, line up

Serial1: HDLC myseq 636127, mineseen 636127, yourseen 515040, line up
Serial1: HDLC myseq 636128, mineseen 636127, yourseen 515041, line up
Serial1: HDLC myseq 636129, mineseen 636129, yourseen 515042, line up

Serial1: HDLC myseq 636130, mineseen 636130, yourseen 515043, line up
Serial1: HDLC myseq 636131, mineseen 636130, yourseen 515044, line up
Serial1: HDLC myseq 636132, mineseen 636130, yourseen 515045, line up
Serial1: HDLC myseq 636133, mineseen 636130, yourseen 515046, line down
Serial1: HDLC myseq 636127, mineseen 636127, yourseen 515040, line up
Serial1: HDLC myseq 636128, mineseen 636127, yourseen 515041, line up
Serial1: HDLC myseq 636129, mineseen 636129, yourseen 515042, line up
```

**1 missed
keepalive** →

**3 missed
keepalives; line
goes down** →

Figure 10-36 Example Debug Serial-Interface Output for HDLC

In Figure 10-36, the **debug serial-interface** display shows that the remote router is not receiving all of the keepalives the router is sending. When the difference in the values in the myseq and mineseen fields exceeds three, the line goes down and the interface is reset.

Table 10-33 describes significant fields shown in Figure 10-36.

Table 10-33 Debug Serial-Interface Field Descriptions for HDLC

Field	Description
Serial1	Interface through which the serial connection is taking place.
HDLC	Indicates that the serial connection is an HDLC connection.
myseq 636119	The myseq counter increases by 1 each time the router sends a keepalive packet to the remote router.
mineseen 636118	The value of the mineseen counter reflects the last myseq sequence number the remote router has acknowledged receiving from the router. The remote router stores this value in its yourseen counter and sends that value in a keepalive packet to the router.
yourseen 515032	The yourseen counter reflects the value of the myseq sequence number the router has received in a keepalive packet from the remote router.
line up	Indicates that the connection between the routers is maintained. Value changes to line down if the values of the myseq and myseen fields in a keepalive packet differ by more than 3. Value returns to line up when the interface is reset. If the line is in loopback mode, (looped) appears after this field.

Table 10-34 describes additional error messages that the **debug serial-interface** command can generate for HDLC.

Table 10-34 Debug Serial-Interface Error Messages for HDLC

Field	Description
Illegal serial link type code xxx, PC = 0xnnnnnnnn	This message is displayed if the router attempts to send a packet containing an unknown packet type.
Illegal HDLC serial type code xxx, PC = 0xnnnnnnnn	This message is displayed if an unknown packet type is received.
Serial 0: attempting to restart	This message is displayed periodically if the interface is down. The hardware is then reset to hopefully correct the problem.
Serial 0: Received bridge packet sent to nnnnnnnnnn	This message is displayed if a bridge packet is received over a serial interface configured for HDLC, and bridging is not configured on that interface.

Debug Serial-Interface for HSSI

On an HSSI interface, the **debug serial-interface** command can generate the following additional error message:

HSSI0: Reset from 0xnnnnnnnnnn

This message indicates that the HSSI hardware has been reset. The 0xnnnnnnnnnn variable is the address of the routine requesting that the hardware be reset; this value is useful only to development engineers.

Debug Serial-Interface for ISDN Basic Rate

Table 10-35 describes error messages that the **debug serial-interface** command can generate for ISDN Basic Rate.

Table 10-35 Debug Serial-Interface Field Descriptions for ISDN Basic Rate

Field	Description
BRI: D-chan collision	Indicates that a collision on the ISDN D channel has occurred; the software will reattempt transmission.
Received SID Loss of Frame Alignment int.	Indicates that the ISDN hardware has lost frame alignment. This usually indicates a problem with the ISDN network.
Unexpected IMP int: ipr = 0xnn	Indicates that the ISDN hardware received an unexpected interrupt. The 0xnn variable indicates the value returned by the interrupt register.
BRI(d): RX Frame Length Violation. Length = n	This message or any of the five messages that follow may be displayed when a receive error occurs on one of the ISDN channels. The (d) indicates which channel. These may indicate a problem with the ISDN network connection.
BRI(d): RX Nonoctet Aligned Frame	
BRI(d): RX Abort Sequence	
BRI(d): RX CRC Error	
BRI(d): RX Overrun Error	
BRI(d): RX Carrier Detect Lost	Indicates that the BRI hardware has been reset. The 0xnnnnnnnn variable is the address of the routine that requested that the hardware be reset; it is useful only to development engineers.
BRI0: Reset from 0xnnnnnnnn	
BRI(d): Bad state in SCMs scm1 = x scm2 = x scm3 = x	This message or any of the two messages that follow are displayed if the ISDN hardware is not in the proper state. The hardware is then reset. If this message is displayed constantly, it usually indicates a hardware problem.
BRI(d): Bad state in SCONs scon1 = x scon2 = x scon3 = x	
BRI(d): Bad state ub SCR; SCR = x	This message is displayed if a packet is received, but the encapsulation used for the packet is not recognized. It can indicate that the interface is misconfigured.
BRI(d): Illegal packet encapsulation = n	

Debug Serial-Interface for an MK5025 Device

Table 10-36 describes the additional error messages that the **debug serial-interface** command can generate for an MK5025 device.

Table 10-36 Debug Serial-Interface Field Descriptions for an MK5025 Device

Field	Description
MK5(d): Reset from 0xnnnnnnnn	This message indicates that the hardware has been reset. The 0xnnnnnnnn variable is the address of the routine that requested that the hardware be reset; it is useful only to development engineers.
MK5(d): Illegal packet encapsulation = <i>n</i>	This message is displayed if a packet is received, but the encapsulation used for the packet is not recognized. Possibly an indication that the interface is misconfigured.
MK5(d): no packet available for packet realignment	This message is displayed in cases where the serial driver attempted to get a buffer (memory) and was unable to do so.
MK5(d): Bad state in CSR0 = (x)	This message is displayed if the hardware is not in the proper state. The hardware is then reset. If this message is displayed constantly, it usually indicates a hardware problem.
MK5(d): New serial state = <i>n</i>	This message is displayed to indicate that the hardware has interrupted the software. It displays the state that the hardware is reporting.
MK5(d): DCD is down. MK5(d): DCD is up.	If the interrupt indicates that the state of carrier has changed, one of these messages is displayed to indicate the current state of DCD.

Debug Serial-Interface for PPP Encapsulation

Figure 10-37 lists all of the messages that the **debug serial-interface** command can generate when the encapsulation is set to PPP and PPP is negotiating configuration options.

```
ppp: deccp_ackci: received bad Ack
ppp: deccp_nakci: received bad Nak
ppp: deccp_rejci: received bad Reject
ppp: ipcp_reqci: bad CI length
ppp: ipcp_ackci: received bad Ack
ppp: ipcp_nakci: received bad Nak
ppp: ipcp_rejci: received bad Reject
ppp: ipcp_reqci: bad CI length
ppp: ipcp_reqci: returning CONFACK
ppp: ipcp_reqci: returning CONFNAK
ppp: ipcp_reqci: returning CONFREJ
ppp: rcvd short code-reject packet
ppp: rcvd code-reject for code n
ppp: received bad configuration ACK
ppp: received bad configuration NAK
ppp: received bad configuration reject
ppp: bad CI length = n
ppp: rcvd unknown option n
```

Figure 10-37 Example Debug Serial-Interface Output for PPP

A knowledge of the PPP protocol is necessary to understand the significance of the messages listed in Figure 10-37.

Figure 10-38 lists the **debug serial-interface** messages that can be displayed when CHAP is enabled on a PPP interface.

```
Attempt to reject authentication ignored.
Serial 0: Unable to respond to CHAP challenge. No USERNAME entry for xxxx
Serial 0: Unable to respond to CHAP challenge. No password defined for \
USERNAME xxx
Serial 0: Failed CHAP authentication with remote.
Serial 0: remote passed CHAP authentication.
Serial 0: Passed CHAP authentication with remote.
```

Figure 10-38 Example Debug Serial-Interface Output When CHAP Is Enabled on a PPP Interface

The messages listed in Figure 10-38 indicate the current state of CHAP negotiation.

Debug Serial-Interface for SMDS Encapsulation

When encapsulation is set to SMDS, **debug serial-interface** displays SMDS packets that have been sent and received, as well as any error messages resulting from SMDS packet transmission.

The error messages that the **debug serial-interface** command can generate for SMDS follow.

The following message indicates that a new protocol requested SMDS to encapsulate the data for transmission. SMDS does not know yet how to encapsulate the protocol.

```
SMDS: Error on Serial 0, encapsulation bad protocol = x
```

The following indicates that SMDS was asked to encapsulate a packet, but no corresponding destination E.164 SMDS address was found in any of the static SMDS tables or in the ARP tables.

```
SMDS send: Error in encapsulation, no hardware address, type = x
```

The following indicates that a protocol such as CLNS or IP has been enabled on an SMDS interface, but the corresponding multicast addresses have not been configured. The *n* variable displays the link type for which encapsulation was requested. This value is only significant to Cisco as an internal protocol type value.

```
SMDS: Send, Error in encapsulation, type=n
```

The following messages can occur when a packet that was somehow corrupted is received on an SMDS interface. The router expected x, but received y.

```
SMDS: Invalid packet, Reserved NOT ZERO, x y
SMDS: Invalid packet, TAG mismatch x y
SMDS: Invalid packet, Bad TRAILER length x y
```

The following messages can indicate an invalid length for an SMDS packet.

```
SMDS: Invalid packet, Bad BA length x
SMDS: Invalid packet, Bad header extension length x
SMDS: Invalid packet, Bad header extensiontype x
SMDS: Invalid packet, Bad header extension value x
```

The following messages are displayed when **debug serial-interface** is enabled.

```
Interface Serial 0 Sending SMDS L3 packet:
SMDS: dgsizex type:0xn src:y dst:z
```

If **debug serial-interface** is enabled, the following message can be displayed when a packet is received on an SMDS interface, but the destination SMDS address does not match any on that interface.

```
SMDS: Packet n, not addressed to us
```

Debug Serial-Packet

Use the **debug serial-packet** command to provide more detailed serial interface debugging information than you can obtain using **debug serial-interface**.

The **debug serial-packet** command generates output that is dependent on the type of serial interface and the encapsulation that is running on that interface. The hardware platform also can impact **debug serial-packet** output.

Currently, the **debug serial-packet** command displays output for only DDR, PPP, and SMDS encapsulations.

Debug Serial-Packet for DDR

When you enable **debug serial-packet** and DDR is enabled on the interface, information concerning the cause of any calls (called Dialing cause) may be displayed.

The following line of output for an IP packet lists the name of the DDR interface and the source and destination addresses of the packet.

```
Dialing cause: Serial0: ip (s=131.108.1.111 d=131.108.2.22)
```

The following line of output for a bridged packet lists the DDR interface and the type of packet (in hexadecimal). For information on these packet types, see Appendix C, "Ethernet Type Codes," of the *Router Products Configuration and Reference* publication.

```
Dialing cause: Serial1: Bridge (0x6005)
```

Debug Serial-Packet for PPP

Figure 10-39 shows example **debug serial-packet** output when PPP is enabled on the interface.

```
ppp: config ACK received, type = nnn
ppp: config ACK received, type = nnn, value = yyy
ppp: config ACK received, type = nnn, value = yyy
ppp: config ACK received, type = nnn
ppp: config ACK received, type = nnn
```

Figure 10-39 Example Debug Serial-Packet Output for PPP

The preceding five messages may appear when PPP is attempting to negotiate a link. They indicate that PPP received an ACK for option type nnn, and if the option has a value, the value that was acked also is displayed. This is possibly useful in debugging PPP link establishment, but is mostly useful with some knowledge of the PPP protocol.

Table 10-37 describes significant fields shown in Figure 10-39.

Table 10-37 Debug Serial-Packet Field Descriptions for PPP

Field	Description
ppp: config ACK received	The router has received an acknowledgment packet in response to the configuration negotiation request packet it sent.
type = nnn	Number indicating the LCP configuration option to be negotiated. Possible values include: <ul style="list-style-type: none"> ■ 1—Maximum-Received-Unit (MRU) ■ 2—Async-Control-Character-Map ■ 3—Authentication-Protocol ■ 4—Quality-Protocol ■ 5—Magic-Number ■ 6—Undefined ■ 7—Protocol-Field-Compression ■ 8—Address-and-Control-Field-Compression ■ 9—32-Bit-FCS
value = yyy	Value of the LCP configuration option that has been negotiated.

Additional messages that the debug serial-packet command can generate when PPP is enabled follow.

The following message is displayed when PPP sends a packet onto the line. The “Serial0” shows the interface the packet is sent on. The state corresponds to a PPP state machine state, and is only useful to Cisco technical support staff. The link can be either ppp-lcp or ppp-ipcp, indicating that it is either a PPP LCP packet or a PPP IPCP packet. The code is the PPP packet type being transmitted, the ID is a sequence number for this packet, and LEN is the length of packet. This message is only displayed for PPP-generated packets, not for all packets using PPP encapsulation.

```
PPP send: on Serial0 STATE= 4 LINK= ppp-lcp, CODE= 5, ID= 345, LEN = 9
```

The following message is displayed if the PPP timer expires. It indicates that the remote side did not respond to the packet in the time allowed.

```
ppp: TIMEout: Time= 3245532 State= 4
```

The following three messages may be displayed if PPP receives a packet that is incorrectly formatted.

```
ppp: rcvd short header for ppp-lcp
ppp: rcvd illegal length for ppp-lcp
ppp: rcvd short packet. len x > y for ppp-lcp
```

The following message is displayed when a PPP-specific packet is received. It either will be for ppp-lcp or ppp-ipcp, depending on which PPP layer the packet is for. It will give a state, which is a state in the PPP state machine; a code, which is the type of PPP packet received; the ID, which is a sequence number; and the length of the packet.

```
PPP input(ppp-lcp): state = 4 code = 5 id = 345 len = 9
```

The following message is displayed if PPP has received an ACK for a configuration request it transmitted. The ID can be matched with an ID displayed in the PPP send debug message to verify which packet was acked.

```
ppp: state = 4 fsm_rconfack(ppp-lcp): rcvd id 345
```

One of the following messages is displayed when PPP receives a configuration packet from the other side. It displays the configuration type and whether there is a value for that type, the value, and whether it is going to ack, nack, or reject this configuration option.

```
ppp: received config for type = x value = y acked
ppp: received config for type = x value = y rejected
ppp: received config for type = x value = y nacked
```

Debug Serial-Packet for SMDS Encapsulation

Figure 10-40 shows example output when SMDS is enabled on the interface.

```
Interface Serial2 Sending SMDS L3 packet:
SMDS Header : Id: 00 RSVD: 00 Bntag: EC Basize: 0044
Dest:E18009999999FFFF Src:C12015804721FFFF Xh:04030000030001000000000000000000
SMDS LLC : AA AA 03 00 00 00 80 38
SMDS Data : E1 19 01 00 00 80 00 00 0C 00 38 1F 00 0A 00 80 00 00 0C 01 2B 71
SMDS Data : 06 01 01 0F 1E 24 00 EC 00 44 00 02 00 00 83 6C 7D 00 00 00 00 00
SMDS Trailer : RSVD: 00 Bntag: EC Length: 0044
```

Figure 10-40 Example Debug Serial-Packet Output for SMDS

As Figure 10-40 suggests, when encapsulation is set to SMDS, **debug serial-packet** displays the entire SMDS header (in hex), as well as some payload data on transmit or receive. This information is useful only when you have an understanding of the SMDS protocol. The first line of the output indicates either Sending or Receiving.

Debug Source-Event

Use the **debug source-event** command to provide informational displays of source bridging activity. (Output of the **debug source-bridge** command is identical to the output of this command.)

Note: In order to use the **debug source-event** command to display traffic source-routed through an interface, you first must disable fast switching of SRB frames with the **no source-bridge route-cache** interface subcommand.

Figure 10-41 shows example **debug source-event** output.

```
RSRB0: forward (srn 5 bn 1 trn 10), src: 8110.2222.33c1 dst: 1000.5a59.04f9 [0800.3201.00A1.0050]
RSRB0: forward (srn 5 bn 1 trn 10), src: 8110.2222.33c1 dst: 1000.5a59.04f9 [0800.3201.00A1.0050]
RSRB0: forward (srn 5 bn 1 trn 10), src: 8110.2222.33c1 dst: 1000.5a59.04f9 [0800.3201.00A1.0050]
RSRB0: forward (srn 5 bn 1 trn 10), src: 8110.2222.33c1 dst: 1000.5a59.04f9 [0800.3201.00A1.0050]
RSRB0: forward (srn 5 bn 1 trn 10), src: 8110.2222.33c1 dst: 1000.5a59.04f9 [0800.3201.00A1.0050]
```

Figure 10-41 Example Debug Source-Event Output

Table 10-38 describes significant fields shown in Figure 10-41.

Table 10-38 Debug Source-Event Field Descriptions

Field	Description
RSRB0:	Indicates that this RIF cache entry is for the Token Ring 0 interface, which has been configured for remote source route bridging. (SRB1, in contrast, would indicate that this RIF cache entry is for Token Ring 1, configured for source route bridging.)
forward	Indicates that this is a forward (normal data) packet, in contrast to a control packet containing proprietary Cisco bridging information.
srn 5	Indicates the ring number of the packet's source ring.
bn 1	Indicates the bridge number of the bridge this packet traverses.
trn 10	Indicates the ring number of the packet's target ring.
src: 8110.2222.33c1	Source address of the route in this RIF cache entry.
dst: 1000.5a59.04f9	Destination address of the route in this RIF cache entry.
[0800.3201.00A1.0050]	RIF string in this RIF cache entry.

Examples of other **debug source-event** messages that can be displayed follow.

In the following example messages, SRB*n* or RSRB*n* denotes a message associated with interface Token Ring *n*. An *n* of 99 denotes the remote side of the network.

```
SRBn: no path, s: <src MAC addr>d: <dst MAC addr>rif: <rif>
```

In the preceding example, a bridgeable packet came in on interface Token Ring *n* but there was nowhere to send it. This is most likely a configuration error. For example, an interface has source bridging turned on, but it is not connected to another source bridging interface or a ring group.

In the following example, a bridgeable packet has been forwarded from Token Ring *n* to the target ring. The two interfaces are directly linked.

```
SRBn: direct forward (srn <ring>brn <bridge>trn <ring>)
```

In the following examples, a proxy explorer reply was not generated because there was no way to get to the address from this interface. The packet came from the node with the first <address>.

```
SRBn: br dropped proxy XID, <address>for <address>, wrong vring (rem)
SRBn: br dropped proxy TEST, <address>for <address>, wrong vring (rem)
SRBn: br dropped proxy XID, <address>for <address>, wrong vring (local)
SRBn: br dropped proxy TEST, <address>for <address>, wrong vring (local)
SRBn: br dropped proxy XID, <address>for <address>, no path
SRBn: br dropped proxy TEST, <address>for <address>, no path
```

In the following example, an appropriate proxy explorer reply was generated on behalf of the second <address>. It is sent to the first <address>.

```
SRBn: br sent proxy XID, <address>for <address>[<rif>]
SRBn: br sent proxy TEST, <address>for <address>[<rif>]
```

The following example indicates that the broadcast bits were not set, or that the routing information indicator on the packet was not set.

```
SRB<unit#>: illegal explorer, s: <srcMACaddr> d: <destMACaddr> rif:
<RIFstring>
```

The following example indicates that the direction bit in the RIF field was set, or that an odd packet length was encountered. Such packets are dropped.

```
SRB<unit #>: bad explorer control, D set or odd
```

The following example indicates that a spanning explorer was dropped because the spanning option was not configured on the interface.

```
SRB<unit #>: span dropped, input off, s: <src mac addr> d: <dest mac addr>
rif: <rif string>
```

The following example indicates that a spanning explorer was dropped because it had traversed the ring previously.

```
SRB<unit #>: span violation, s: <src mac addr> d: <dest mac addr> rif:
<rif string>
```

The following example indicates that an explorer was dropped because the maximum hop count limit was reached on that interface.

```
SRB<unit #>: max hops reached - <hop cnt>, s: <src mac addr> d: <dest mac addr>
rif: <rif string>
```

The following example indicates that the ring exchange request was sent to the indicated peer. This request tells the remote side which rings this node has and asks for a reply indicating which rings that side has.

```
RSRB: sent RingXreq to <ring group>/<ip addr>
```

The following example indicates that a message has been sent to the remote peer. The <label> variable can be AHDR (active header), PHDR (passive header), HDR (normal header), or DATA (data exchange), and <op> can be Forward, Explorer, Ring Xchg, Req, Ring Xchg, Rep, Unknown Ring Group, Unknown Peer, or Unknown Target Ring.

```
RSRB: <label>: sent <op>to <ring group>/<ip addr>
```

The following example indicates that the remote bridge and ring pair have been removed from or added to the local ring group table because the remote peer has changed.

```
RSRB: removing bn <bridge>rn <ring>from <ring group>/<ip addr>  
RSRB: added bridge <bridge>, ring <ring>for <ring group>/<ip addr>
```

The following example shows miscellaneous remote peer connection establishment messages.

```
RSRB: peer <ring group>/<ip addr>closed [last state n  
RSRB: passive open <ip addr>(remote port) -><local port>  
RSRB: CONN: opening peer <ring group>/<ip addr>, attempt n  
RSRB: CONN: Remote closed <ring group>/<ip addr>on open  
RSRB: CONN: peer <ring group>/<ip addr>open failed, <reason>[code]
```

The following example shows that an explorer packet was propagated onto the local ring from the remote ring group.

```
RSRBn: sent local explorer, bridge <bridge>trn <ring>, [rif]
```

The following messages indicate that the remote source-route bridging code found the packet to be in error.

```
RSRBn: ring group <ring group>not found  
RSRBn: explorer rif [rif] not long enough
```

The following example indicates that a buffer could not be obtained for a Ring Exchange Packet; this is an internal error.

```
RSRB: couldn't get pak for ringXchg
```

The following example indicates that a ring exchange packet was received that had an incorrect length; this is an internal error.

```
RSRB: XCHG: req/reply badly formed, length <pak length>, peer <peer id>
```

The following example indicates that a ring entry was removed for the peer; the ring was possibly disconnected from the network, causing the remote router to send an update to all its peers.

```
RSRB: removing bridge <br #> ring <ring #> from <peer name> <ring type>
```


The following example indicates that a ring entry was added for the specified peer; the ring was possibly added to the network, causing the other router to send an update to all its peers.

```
RSRB: added bridge <br #>, ring <ring #> for <peer id>
```

The following example indicates that no memory was available to add a ring number to the ring group specified; this is an internal error.

```
RSRB: no memory for ring element <ring group #>
```

The following example indicates that memory was corrupted for a connection block; this is an internal error.

```
RSRB: CONN: corrupt connection block
```

The following example indicates that a connector process started, but that there was no packet to process; this is an internal error.

```
RSRB: CONN: warning, no initial packet, peer: <ip addr> <peer pointer>
```

The following example indicates that a packet was received with a version number different from the one present on the router.

```
RSRB: IF New version. local=<local version #>, remote=<remote version>,  
<pak op code> <peer id>
```

The following example indicates that a packet with a bad op code was received for a direct encapsulation peer; this is an internal error.

```
RSRB: IFin: bad op <op code> (op code string) from <peer id>
```

The following example indicates that the virtual ring header will not fit on the packet to be sent to the peer; this is an internal error.

```
RSRB: vrif_sender, hdr won't fit
```

The following example indicates that the specified peer is being opened. The retry count specifies the number of times the opening operation is attempted.

```
RSRB: CONN: opening peer <peer id> <retry count>
```

The following example indicates that the router, configured for FST encapsulation, received a version reply to the version request packet it had sent previously.

```
RSRB: FST Rcvd version reply from <peer id> (version #)
```

The following example indicates that the router, configured for FST encapsulation, sent a version request packet to the specified peer.

```
RSRB: FST Version Request. op = <opcode>, <peer id>
```

The following example indicates that the router received a packet with a bad op code from the specified peer; this is an internal error.

```
RSRB: FSTin: bad op <opcode> (op code string) from <peer id>
```

The following example indicates that the TCP connection between the router and the specified peer is being aborted.

```
RSRB: aborting <ring group #>/<peer id> (vrtcpd_abort called)
```

The following example indicates that an attempt to establish a TCP connection to a remote peer timed out.

```
RSRB: CONN: attempt timed out
```

The following example indicates that a packet was dropped because the ring group number in the packet did not correlate with the ring groups configured on the router.

```
RSRB<unit #>: ring group <ring group #> not found
```

Debug Span

Use the **debug span** command to track changes in the spanning-tree topology when debugging a transparent bridge.

This command is useful for verifying that the spanning-tree protocol is operating correctly.

IEEE Spanning Tree Example

Example **debug span** output for an IEEE BPDU packet follows:

ST: Ether4 0000000000000A080002A02D6700000000000A080002A02D6780010000140002000F00

Broken up by fields and labeled to aid documentation, the preceding **debug span** output appears as shown in Figure 10-42.

ST: Ether4 0000 00 00 00 000A 080002A02D67 00000000 000A 080002A02D67 80 01 0000 1400 0200 0F00
 A B C D E F G H I J K L M N O

Figure 10-42 Example Debug Span Output

Table 10-39 describes significant fields shown in this **debug span** output.

Table 10-39 Debug Span Field Descriptions for an IEEE BPDU Packet

Field	Description
ST:	Indicates that this is a spanning tree packet
Ethernet4	Interface receiving the packet
(A) 0000	Indicates that this is an IEEE BPDU packet
(B) 00	Version
(C) 00	Type <ul style="list-style-type: none"> ■ 00 indicates config BPDU ■ 80 indicates TCN BPDU
(D) 00	Topology change acknowledgement <ul style="list-style-type: none"> ■ 00 indicates no change ■ 80 indicates a change notification
(E) 000A	Root priority
(F) 080002A02D67	Root ID
(G) 00000000	Root path cost (0 means the sender of this BPDU packet is the root bridge)
(H) 000A	Bridge priority
(I) 080002A02D67	Bridge ID
(J) 80	Port priority
(K) 01	Port #1
(L) 0000	Message age (in 256ths of a second)
(M) 1400	Maximum age (in 256ths of a second)
(N) 0200	Hello time (in 256ths of a second)
(O) 0F00	Forward delay (in 256ths of a second)

DEC Spanning Tree Example

Example **debug span** output for a DEC BPDU packet follows:

```
ST: Ethernet4 E1190100000200000C01A2C90064008000000C0106CE0A01050F1E6A
```

Broken up by fields and labeled to aid documentation, this **debug span** output appears as shown in Figure 10-43.

```
E1 19 01 00 0002 00000C01A2C9 0064 0080 00000C0106CE 0A 01 05 0F 1E 6A
A  B C  D  E          F      G  H          I      J  K  L  M  N  O
```

Figure 10-43 Example Debug Span Output

Table 10-40 describes significant fields shown in this **debug span** output.

Table 10-40 Debug Span Field Descriptions for a DEC BPDU Packet

Field	Description
ST:	Indicates that this is a spanning tree packet
Ethernet4	Interface receiving the packet
(A) E1	Indicates that this is a DEC BPDU packet
(B) 19	Indicates that this is a DEC Hello packet <ul style="list-style-type: none"> ■ 0x19—DEC Hello ■ 0x02—Topology change notification (TCN)
(C) 01	DEC version
(D) 00	Flag which is a bit field with the following mapping: <ul style="list-style-type: none"> 1—TCN 2—TCN acknowledgment 8—Use short timers
(E) 0002	Root priority
(F) 00000C01A2C9	Root ID (MAC address)
(G) 0064	Root path cost (translated as 100 in decimal notation)
(H) 0080	Bridge priority
(I) 00000C0106CE	Bridge ID
(J) 0A	Port ID (in contrast to interface number)
(K) 01	Message age (in seconds)
(L) 05	Hello time (in seconds)
(M) 0F	Maximum age (in seconds)
(N) 1E	Forward delay (in seconds)
(O) 6A	Not applicable

Debug TFTP

Use the **debug tftp** command to display TFTP debugging information when encountering problems netbooting or using the **configure network** or **write network** commands.

Figure 10-45 shows example **debug tftp** output from the EXEC command **write network**.

```
TFTP: msclock 0x292B4; Sending  write request (retry 0), socket_id 0x301DA8
TFTP: msclock 0x2A63C; Sending  write request (retry 1), socket_id 0x301DA8
TFTP: msclock 0x2A6DC; Received ACK for block 0,  socket_id 0x301DA8
TFTP: msclock 0x2A6DC; Received ACK for block 0,  socket_id 0x301DA8
TFTP: msclock 0x2A6DC; Sending  block 1 (retry 0), socket_id 0x301DA8
TFTP: msclock 0x2A6E4; Received ACK for block 1,  socket_id 0x301DA8
```

Figure 10-44 Example Debug TFTP Output

Table 10-44 describes significant fields shown in the first line of output from Figure 10-45.

Table 10-41 Debug TFTP Field Descriptions

Message	Description
TFTP:	Indicates that this entry describes a TFTP packet.
msclock 0x292B4;	Internal timekeeping clock (in milliseconds).
Sending write request (retry 0)	Indicates the TFTP operation.
socket_id 0x301DA8	Unique memory address for the socket for the TFTP connection.

Debug Token-Ring

Use the **debug token-ring** command to display messages about Token Ring interface activity. This command reports several lines of information for each packet sent or received and is intended for low traffic, detailed debugging.

The Token Ring interface records provide information regarding the current state of the ring. These messages are only displayed when **debug token-events** is enabled.

The **debug token-ring** command invokes verbose Token Ring hardware debugging. This includes detailed displays as traffic arrives and departs the unit.

Note: It is best to use this command only on router/bridges with light loads.

Figure 10-45 shows example **debug token-ring** output.

```
TR0: Interface is alive, phys. addr 5000.1234.5678
TR0: in: MAC: acfc: 0x1105 Dst: c000.ffff.ffff Src: 5000.1234.5678 bf: 0x45
TR0: in:      riflen 0, rd_offset 0, llc_offset 40
TR0: out: MAC: acfc: 0x0040 Dst: 5000.1234.5678 Src: 5000.1234.5678 bf: 0x00
TR0: out: LLC: AAAA0300 00009000 00000100 AAC00000 00000802 50001234 ln: 28
TR0: in: MAC: acfc: 0x1140 Dst: 5000.1234.5678 Src: 5000.1234.5678 bf: 0x09
TR0: in: LLC: AAAA0300 00009000 00000100 AAC0B24A 4B4A6768 74732072 ln: 28
TR0: in:      riflen 0, rd_offset 0, llc_offset 14
TR0: out: MAC: acfc: 0x0040 Dst: 5000.1234.5678 Src: 5000.1234.5678 bf: 0x00
TR0: out: LLC: AAAA0300 00009000 00000100 D1D00000 FE11E636 96884006 ln: 28
TR0: in: MAC: acfc: 0x1140 Dst: 5000.1234.5678 Src: 5000.1234.5678 bf: 0x09
TR0: in: LLC: AAAA0300 00009000 00000100 D1D0774C 4DC2078B 3D000160 ln: 28
TR0: in:      riflen 0, rd_offset 0, llc_offset 14
TR0: out: MAC: acfc: 0x0040 Dst: 5000.1234.5678 Src: 5000.1234.5678 bf: 0x00
TR0: out: LLC: AAAA0300 00009000 00000100 F8E00000 FE11E636 96884006 ln: 28
```

Figure 10-45 Example Debug Token-Ring Output

Descriptions of example lines of output in Figure 10-45 follow.

Table 10-44 describes significant fields shown in the second line of output from Figure 10-45.

```
TR0: in: MAC: acfc: 0x1105 Dst: c000.ffff.ffff Src: 5000.1234.5678 bf: 0x45
```

Table 10-42 Debug Token Ring Field Descriptions—Part 1

Message	Description
TR0:	Name of the interface associated with the Token Ring event.
in:	Indicates whether the packet was input to the interface (in) or output from the interface (out).
MAC:	Indicates the type of packet, as follows: <ul style="list-style-type: none"> ■ MAC—Media Access Control ■ LLC—Link Level Control
acfc: 0x1105	Access Control, Frame Control bytes, as defined by the IEEE 802.5 standard.
Dst: c000.ffff.ffff	Destination address of the frame.
Src: 5000.1234.5678	Source address of the frame.
bf: 0x45	Bridge flags for internal use by technical support staff.

Table 10-44 describes significant fields shown in the third line of output from Figure 10-45.

```
TR0: in:          riflen 0, rd_offset 0, llc_offset 40
```

Table 10-43 Debug Token Ring Field Descriptions—Part 2

Message	Description
TR0:	Name of the interface associated with the Token Ring event.
in:	Indicates whether the packet was input to the interface (in) or output from the interface (out).
riflen 0	Length of the RIF field (in bytes)
rd_offset 0	Offset (in bytes) of the frame pointing to the start of the RIF field.
llc_offset 40	Offset in the frame pointing to the start of the llc field

Table 10-44 describes significant fields shown in the fifth line of output from Figure 10-45.

```
TR0: out: LLC: AAAA0300 00009000 00000100 AAC00000 00000802 50001234 ln: 28
```

Table 10-44 Debug Token Ring Field Descriptions—Part 3

Message	Description
TR0:	Name of the interface associated with the Token Ring event.
out:	Indicates whether the packet was input to the interface (in) or output from the interface (out).
LLC:	Indicates the type of frame, as follows: MAC—Media Access Control LLC—Link Level Control
AAAA0300....	This and the octets that follow it indicate the contents (hex) of the frame.
ln: 28	Indicates the length of the information field (in bytes).

Debug VINES-ARP

Use the **debug vines-arp** command to enable logging of all ARP packets that the router sends or receives.

Figure 10-46 shows example **debug vines-arp** output.

```
VINES: received ARP type 0 from 0260.8c43.a7e4
VINES: sending ARP type 1 to 0260.8c43.a7e4
VINES: received ARP type 2 from 0260.8c43.a7e4
VINES: sending ARP type 3 to 0260.8c43.a7e4 assigning address 3001153C:8004
```

Figure 10-46 Example Debug VINES-ARP Output

In Figure 10-46, the first line shows that the router received an ARP request (type 0) from station address 0260.8c43.a7e4. The second line shows that the router is sending back the ARP service response indicating that it is willing to assign VINES internet addresses. The third line shows that the router received a Vines internet address assignment request (type 2) from address 0260.8c43.a7e4. The fourth line shows that the router is responding (type 3) to the address assignment request from the client and assigning it the address 3001153C:8004.

Table 10-45 describes significant fields shown in Figure 10-46.

Table 10-45 Debug VINES-ARP Field Descriptions

Field	Description
VINES:	Indicates that this is one of the Banyan VINES debugging messages.
received ARP type 0	Indicates that an ARP request of type 0 was received. Possible type values follow: <ul style="list-style-type: none">■ 0—Query request. The ARP client broadcasts a type 0 message to request an ARP service to respond.■ 1—Service response. The ARP service responds with a type 1 message to an ARP client's Query request.■ 2—Assignment request. The ARP client responds to a service response with a type 2 message to request a Banyan VINES Internet address.■ 3—Assignment response. The ARP service responds to an assignment request with a type 3 message that includes the assigned Banyan VINES Internet address.
from 0260.8c43.a7e4	Indicates the source address of the packet.

Debug VINES-Echo

Use the **debug vines-echo** command to enable logging of all MAC-level echo packets that the router sends or receives. Banyan interface testing programs make use of these echo packets.

Note: These echo packets do not include network layer addresses.

Figure 10-47 shows example **debug vines-echo** output.

```
VINESECHO: 100 byte packet from 0260.8c43.a7e4
```

Figure 10-47 Example Debug VINES-Echo Output

Table 10-46 describes the fields shown in Figure 10-47.

Table 10-46 Debug VINES-Echo Field Descriptions

Field	Description
VINESECHO	Indicates that this is a debug vines-echo message.
100 byte packet	Packet size in bytes.
from 0260.8c43.a7e4	Source address of the echo packet.

Debug VINES-Packet

Use the **debug vines-packet** command to enable logging of general VINES debugging information. This information includes packets received, generated, and forwarded, as well as failed access checks and other items.

Figure 10-48 shows example **debug vines-packet** output.

```
VINES: s=30028CF9:1 (Ether2), d=FFFFFFFF:FFFF, rcvd w/ hops 0
VINES: s=3000CBD4:1 (Ether1), d=3002ABEA:1 (Ether2), g=3002ABEA:1, sent
VINES: s=3000CBD4:1 (Ether1), d=3000B959:1, rcvd by gw
VINES: received vines IPC Disc from 3000CBD4:1
VINES: s=3000B959:1 (local), d=3000CBD4:1 (Ether1), g=3000CBD4:1, sent
```

Figure 10-48 Example Debug VINES-Packet Output

The following describes selected lines of output from Figure 10-48.

Table 10-47 describes the fields shown in the first line of output.

Table 10-47 Debug VINES-Packet Field Descriptions

Field	Description
VINES:	Indicates that this is a Banyan VINES packet.
s = 30028CF9:1	Source address of the packet.
(Ether2)	Interface associated with this address.
d = FFFFFFFFF:FFFF	Indicates that the destination is a broadcast address.
rcvd w/ hops 0	Indicates that the packet was received because it was a broadcast packet. The remaining hop count in the packet was zero (0).

In the second line of output that follows, the destination is the address 3002ABEA:1 associated with interface Ether2. Source address 3000CBD4:1 sent a packet to this destination through the gateway at address 3000ABEA:1.

```
VINES: s=3000CBD4:1 (Ether1), d=3002ABEA:1 (Ethernet2), g=3002ABEA:1, sent
```

In the third line of output that follows, the router being debugged is the destination address (3000B959:1).

```
VINES: s=3000CBD4:1 (Ether1), d=3000B959:1, rcvd by gw
```

The following fourth line of output indicates that the packet described in the preceding line was a VINES IPC Disconnect packet.

```
VINES: received vines IPC Disc from 3000CBD4:1
```

In the following fifth line of output, (local) indicates that the router being debugged is the source address for the packet.

```
VINES: s=3000B959:1 (local), d=3000CBD4:1 (Ether1), g=3000CBD4:1, sent
```

Debug VINES-Routing

Use the **debug vines-routing** command to enable logging of all RTP update messages sent or received and all routing table activities that occur in the router.

Figure 10-49 shows example **debug vines-routing** output.

```
Update sent → VINESRTP: sending update to FFFFFFFF:FFFF on Ethernet3
               network 3000073B, metric 2 (.4 seconds)
               network 27AF9A, metric 2 (0.4 seconds)
Update received → VINESRTP: received update from 27AF9A:1 on Ethernet2
                  network 27AF9A from the server
                  network 30019AC7, metric 2 (0.4 seconds)
                  network 3002ABEA, metric 2 (0.4 seconds)
```

Figure 10-49 Example Debug VINES-Routing Output

Figure 10-49 describes two VINES routing updates; the first includes two entries and the second includes three entries. The following describes selected lines of output from Figure 10-49.

The following first line shows that the router sent a periodic routing update to the broadcast address FFFFFFFF:FFFF through the Ethernet3 interface.

```
VINESRTP: sending update to FFFFFFFF:FFFF on Ethernet3
```

The following second line indicates that the router knows how to reach network 3000073B, which is metric 2 away from the router. The value that follows the metric (0.4 seconds) interprets the metric in seconds.

```
network 3000073B, metric 2 (.4 seconds)
```

The following fourth line of output indicates that the router received a routing update from the VINES server at VINES address 27AF9A:1 through the Ethernet2 interface.

```
VINESRTP: received update from 27AF9A:1 on Ethernet2
```

The following fifth line of output implies that the server sending this update is directly accessible to the router (even though VINES servers do not explicitly list themselves in routing updates). Because this is an implicit entry in the table, there is no metric associated with this line of output.

```
network 27AF9A from the server
```

As the first actual entry in the routing update from the VINES server at 27AF9A:1, the following line indicates that network 30019AC7 can be reached by sending to this server. This network is a metric of 2 away from the sending server. The value that follows the metric (0.4 seconds) interprets the metric in seconds.

```
network 30019AC7, metric 2 (0.4 seconds)
```

Debug VINES-Table

Use the **debug vines-table** command to enable logging of all modifications to the VINES routing table. This command provides a subset of the information provided by the **debug vines-routing** command, as well as some more detailed information on table additions and deletions.

Figure 10-50 shows example **debug vines-table** output.

```
VINESRTP: create neighbor 3001153C:8004, interface Ethernet0
```

Figure 10-50 Example Debug VINES-Table Output

Table 10-48 describes significant fields shown in Figure 10-50.

Table 10-48 Debug VINES-Table Field Descriptions

Field	Description
VINESRTP:	Indicates that this is a debug vines-routing or debug vines-table message.
create neighbor 3001153C:8004	Indicates that the client at address 3001153C:8004 has been added to the Banyan VINES neighbor table.
interface Ethernet 0	Indicates that this particular neighbor can be reached through the router interface named Ethernet0.

Debug XNS-Packet

Use the **debug xns-packet** command to enable logging of XNS packet traffic, including the addresses for source, destination, and next hop router of each packet.

Note: To gain the fullest understanding of XNS routing activity, you should enable **debug xns-routing** and **debug xns-packet** together.

Figure 10-51 shows example **debug xns-packet** output.

```
XNS: src=5.0000.0c02.6d04, dst=5.ffff.ffff.ffff, packet sent
XNS: src=1.0000.0c00.440f, dst=1.ffff.ffff.ffff, rcvd. on Ethernet0
XNS: src=1.0000.0c00.440f, dst=1.ffff.ffff.ffff, local processing
```

Figure 10-51 Example Debug XNS-Packet Output.

Table 10-49 describes significant fields shown in Figure 10-51.

Table 10-49 Debug XNS-Packet Field Descriptions

Field	Description
XNS:	Indicates that this is an XNS packet.
src = 5.0000.0c02.6d04	Indicates that the source address for this message is 0000.0c02.6d04 on network 5.
dst = 5.ffff.ffff.ffff	Indicates that the destination address for this message is the broadcast address <code>ffff.ffff.ffff</code> on network 5.
packet sent	Indicates that the packet to destination address <code>5.ffff.ffff.ffff</code> in Figure 10-51, as displayed using the debug xns-packet command, was queued on the output interface.
rcvd. on Ethernet0	Indicates that the router just received this packet through the Ethernet0 interface.
local processing	Indicates that the router has examined the packet and determined that it must process it, rather than forwarding it.

Debug XNS-Routing

Use the **debug xns-routing** command to display XNS routing transactions.

Figure 10-52 shows example **debug xns-routing** output.

```
XNSRIP: sending standard periodic update to 5.ffff.ffff.ffff via Ethernet2
network 1, hop count 1
network 2, hop count 2

XNSRIP: got standard update from 1.0000.0c00.440f socket 1 via Ethernet0
net 2: 1 hops
```

Figure 10-52 Example Debug XNS-Routing Output

Table 10-50 describes significant fields shown in Figure 10-52.

Table 10-50 Debug XNS-Routing Field Descriptions

Field	Description
XNSRIP:	Indicates that this is an XNS routing packet.
sending standard periodic update to 5.ffff.ffff.ffff	The router indicates that this is a periodic XNS routing information update.
via Ethernet2	Indicates that the destination address is ffff.ffff.ffff on network 5.
network 1, hop count 1	Name of the output interface.
got standard update from 1.0000.0c00.440f	Indicates that network 1 is one hop away from this router.
socket 1	The router indicates that it has received an XNS routing information update from address 0000.0c00.440f on network 1.
	The socket number is a well-known port for XNS. Possible values include:
	■ 1—routing information
	■ 2—echo
	■ 3—router error

Debug X25

Use the **debug x25** command to display all X.25 traffic.

This command is particularly useful for diagnosing problems encountered when placing CALLs.

While **debug x25** output includes both data and control messages for all of the router's virtual circuits, **debug x25-events** output (discussed later in this chapter) includes only control messages for all of the router's VCs. In contrast, **debug x25-vc** output (also discussed later) includes only control messages for a particular VC. Thus, **debug x25-events** output is a subset of **debug x25** output, and **debug x25-vc** output is a subset of **debug x25-events** output.



Caution: Because **debug x25** displays all X.25 traffic, it is very processor intensive and can render the router useless. You should only use **debug x25** when the aggregate of all X.25 traffic is less than five packets per second.

Figure 10-53 shows example **debug x25** output.

```
Serial2 (236414440): X25 O R3 RESTART (5) 8 lci 0 cause 7 diag 0
Serial2 (236414444): X25 I R3 RESTART (5) 8 lci 0 cause 0 diag 0
Serial2 (236424436): X25 I P1 CALL REQUEST (11) 8 lci 1024
From(2): 49 To(2): 46
Facilities: (0)
Serial2 (236424436): X25 first byte of call user data (4): 0xCC
Serial2 (236424440): X25 O P4 CALL CONNECTED (3) 8 lci 1024
Serial2 (236426444): X25 I P4 DATA (103) 8 lci 1024 PS 0 PR 0
Serial2 (236426448): X25 O D1 DATA (103) 8 lci 1024 PS 0 PR 1
Serial2 (236426460): X25 I D1 DATA (103) 8 lci 1024 PS 1 PR 0
Serial2 (236426464): X25 O D1 DATA (103) 8 lci 1024 PS 1 PR 2
Serial2 (236426484): X25 I D1 RR (3) 8 lci 1024 PR 2
Serial2 (236426500): X25 I D1 DATA (103) 8 lci 1024 PS 2 PR 2
Serial2 (236426500): X25 O D1 DATA (103) 8 lci 1024 PS 2 PR 3
Serial2 (236453060): X25 I D1 CLEAR REQUEST (5) 8 lci 1024 cause 0 diag 122
Serial2 (236453060): X25 O D1 CLEAR CONFIRMATION (3) 8 lci 1024
Serial2 (236453064): X25 I D1 RESET REQUEST (5) 8 lci 1 cause 0 diag 122
Serial2 (236453064): X25 O D1 RESET CONFIRMATION (3) 8 lci 1
```

Figure 10-53 Example Debug X25 Output

Figure 10-53 shows a typical exchange of packets between two X.25 devices on a network. The first line of output in Figure 10-53, reproduced below, describes a RESTART packet.

```
Serial2 (236414440): X25 O R3 RESTART (5) 8 lci 0 cause 7 diag 0
```

Table 10-51 describes the fields shown in this line of output.

Table 10-51 Debug X25 Field Descriptions

Field	Description
Serial2 (236426440)	Indicates the interface associated with this X.25 event. System clock (in milliseconds). Useful for determining the amount of time between events.
X25	Indicates that this message describes an X.25 event.
O	Indicates whether the X.25 message was input (I) or output (O) through the interface.
R3	State of the virtual circuit. Possible values follow. <ul style="list-style-type: none"> ■ D1—Flow control ready ■ D2—DTE reset request ■ D3—DCE reset indication ■ P1—Idle ■ P2—DTE waiting for DCE to connect CALL ■ P3—DCE waiting for DTE to accept 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 ■ X1—Nonstandard state for a virtual circuit in hold-down <p>See Annex B of the 1988 CCITT X.25 Recommendation for more information on these states.</p>

Field	Description
RESTART	Describes the type of X.25 packet. Possible values follow. <ul style="list-style-type: none"> ■ CALL CONNECTED ■ CALL REQUEST ■ CLEAR CONFIRMATION ■ CLEAR REQUEST ■ DATA ■ DIAGNOSTIC ■ ILLEGAL ■ INTR CONFIRMATION ■ INTR (interrupt) ■ REGISTRATION ■ REGISTRATION CONFIRMATION ■ RESET CONFIRMATION ■ RESET REQUEST ■ RESTART ■ RESTART CONFIRMATION ■ RNR (Receiver Not Ready) ■ RR (Receiver Ready)
(5)	Number of bytes in the packet.
8	Modulo of the virtual circuit. Possible values are 8 or 128.
lci 0	Virtual circuit number. See Annex A of the 1988 CCITT X.25 Recommendation for information on VC assignment.
cause 7	Code indicating the event that triggered the packet. The cause field can only appear in entries for CLEAR REQUEST, DIAGNOSTIC, RESET REQUEST, and RESTART packets. Possible values for the cause field can vary, depending on the type of packet. Refer to Appendix A of this manual, "X.25 Cause and Diagnostics Codes," for explanations of these codes.
diag 0	Code providing an additional hint as to what, if anything, went wrong. The diag field can only appear in entries for CLEAR REQUEST, RESET REQUEST and RESTART packets. Because of the large number of possible values, they are listed in Appendix A of this manual, "X.25 Cause and Diagnostics Codes."

Notice that the first DATA packet in Figure 10-53 contains two fields not yet documented.

```
Serial2 (236426444): X25 I P4 DATA (103) 8 lci 1024 PS 0 PR 0
```

Table 10-52 describes the PS and PR fields that can appear in a debug x25 display.

Table 10-52 Debug X25 PS and PR Field Descriptions

Field	Description
PS 7	Packet send sequence number; used for flow control of the sending packet. Present only in DATA packets.
PR 5	Packet receive sequence number; used for flow control of the receiving packet. Present only in DATA, RR and RNR packets.

In Figure 10-53, notice also that the CALL REQUEST packet precedes two other lines of output that are unique in format.

```
Serial2 (236424436): X25 I P1 CALL REQUEST (11) 8 lci 1024
From(2): 49 To(2): 46
Facilities: (0)
```

These lines indicate that the CALL REQUEST packet has a two-digit source address, 49, and a two-digit destination address, 46. These are X.121 addresses that can be from 0 to 15 digits in length. The Facilities field is (0) bytes in length, indicating that no X.25 facilities are being requested.

Debug X25-Events

Use the **debug x25-events** command to display all X.25 traffic except X.25 data or acknowledgment packets.

The **debug x25-events** command is very useful for debugging X.25 problems, because it shows changes that occur in the virtual circuits handled by the router. Because most X.25 connectivity problems stem from errors that CLEAR or RESET virtual circuits, you can use **debug x25-events** to identify these errors.

While **debug x25** output includes both data and control messages for all of the router's virtual circuits, **debug x25-events** output includes only control messages for all of the router's VCs. In contrast, **debug x25-vc** output includes only control messages for a particular VC. Thus, **debug x25-events** output is a subset of **debug x25** output, and **debug x25-vc** output is a subset of **debug x25-events** output.

Note: Because **debug x25-events** displays a subset of all X.25 traffic, it is safer to use during production hours.

Figure 10-54 shows example **debug x25-events** output.

```
Serial2 (236543524): X25 O R3 RESTART (5) 8 lci 0 cause 7 diag 0
Serial2 (236543528): X25 I R3 RESTART (5) 8 lci 0 cause 0 diag 0
Serial2 (236552660): X25 I P1 CALL REQUEST (11) 8 lci 1024
From(2): 49 To(2): 46
Facilities: (0)
Serial2 (236552660): X25 first byte of call user data (4): 0xCC
Serial2 (236552664): X25 O P4 CALL CONNECTED (3) 8 lci 1024
Serial2 (236564056): X25 I D1 CLEAR REQUEST (5) 8 lci 1024 cause 0 diag 122
Serial2 (236564056): X25 O D1 CLEAR CONFIRMATION (3) 8 lci 1024
Serial2 (236564060): X25 I D1 RESET REQUEST (5) 8 lci 1 cause 0 diag 122
Serial2 (236564060): X25 O D1 RESET CONFIRMATION (3) 8 lci 1
```

Figure 10-54 Example Debug X25-Events Output

See the Debug X.25 command description for information on the fields in **debug x25-events** output.

Debug X25-VC

Use the **debug x25-vc** command to display traffic for a particular virtual circuit in order to solve any connectivity or performance problems it is exhibiting.

Syntax for the **debug x25-vc** command follows.

debug x25-vc *number*

In this syntax statement, *number* is the LCI number associated with the virtual circuit(s) you want to monitor. Because no interface is specified, traffic on any VC that has the specified *number* is reported.

While **debug x25** output includes both data and control messages for all of the router's virtual circuits, **debug x25-events** output includes only control messages for all of the router's VCs. In contrast, **debug x25-vc** output includes only control messages for a particular VC. Thus, **debug x25-events** output is a subset of **debug x25** output, and **debug x25-vc** output is a subset of **debug x25-events** output.

Note: Because **debug x25-vc** only displays traffic for a small subset of virtual circuits, it is safe to use even under heavy traffic conditions, as long as events for that virtual circuit are fewer than 25 packets per second.

Figure 10-55 shows example **debug x25-vc** output.

```
Serial0: X25 I R1 RESTART (5) 8 lci 0 cause 7 diag 250
Serial0: X25 0 R1 RESTART CONFIRMATION (3) 8 lci 0
Serial0: X25 0 P2 CALL REQUEST (19) 8 lci 1
From(14): 31250000000101 To(14): 31109090096101
Facilities (0)
Serial0: X25 0 P6 CLEAR REQUEST (5) 8 lci 1 cause diag 122
```

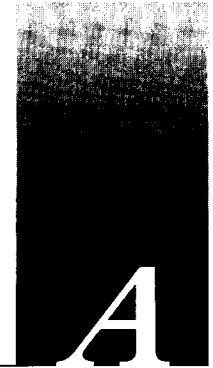
Figure 10-55 Example Debug X25-VC Output

See the Debug X.25 command description for information on the fields in **debug x25-events** output.



Appendix A

X.25 Cause and Diagnostic Codes



This appendix covers the X.25 cause and diagnostics codes, as referred to in the Debug X.25 section of Chapter 10, “Debug Command Reference.” For more information on these codes, see the 1988 CCITT X.25 Recommendation.

Note: The router reports the decimal value of a cause or diagnostic code, whereas other X.25 equipment may report these codes in other values such as hexadecimal. For this reason, this appendix lists both the decimal and hexadecimal values of the cause and diagnostic codes.

X.25 Cause Codes

A cause code indicates an event that triggered an X.25 packet. The cause code can only appear in entries for CLEAR REQUEST, REGISTRATION CONFIRMATION, RESET REQUEST, and RESTART packets. Possible values for the cause code can vary, depending on the type of packet. Because the REGISTRATION exchange is not supported, those cause codes are not documented in this section.

Table A-2 describes the meanings of cause codes for CLEAR REQUEST packets.

Table A-1 Cause Code Descriptions for CLEAR REQUEST Packets

Code (Hex)	Code (Dec)	Description
00	0 (or 128 to 255)	DTE originating
01	1	Number busy
03	3	Invalid facility request
05	5	Network congestion
09	9	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 not accepted
21	33	Incompatible destination
29	41	Fast select not accepted
39	57	Ship absent

Table A-3 describes the meanings of cause codes for RESET REQUEST packets.

Table A-2 Cause Code Descriptions for RESET REQUEST Packets

Code (Hex)	Code (Dec)	Description
00	0 (or 128 to 255)	DTE originated
01	1	Out of order
03	3	Remote procedure error
05	5	Local procedure error
07	7	Network congestion
09	9	Remote DTE operational
0F	15	Network operational
11	17	Incompatible destination
1D	29	Network out of order

Table A-3 describes the meanings of cause codes for RESTART packets.

Table A-3 Cause Code Descriptions for RESTART Packets

Code (Hex)	Code (Dec)	Description
00	0 (or 128 to 255)	DTE restarting
01	1	Local procedure error
03	3	Network congestion
07	7	Network operational
7F	127	Registration/cancellation confirmed

X.25 Diagnostic Codes

The diag (diagnostic) code provides an additional hint as to what, if anything, went wrong. This code can only appear in entries for CLEAR REQUEST, DIAGNOSTIC, RESET REQUEST, and RESTART packets. Unlike the cause codes, the diag codes do not vary depending upon the type of packet.

Note: These diagnostic codes can be produced by any equipment handling a given virtual circuit, and are then propagated through all equipment handling that virtual circuit. Thus, receipt of a diagnostic code may not indicate a problem with the router at all.

Table A-4 describes the meanings of possible diag codes.

Table A-4 Diagnostic Field Code Descriptions

Code (Hex)	Code (Dec)	Description
00	00	No additional information
01	01	Invalid P(S)
02	02	Invalid P(R)
10	16	Packet type invalid
11	17	Packet type invalid for state R1
12	18	Packet type invalid for state R2
13	19	Packet type invalid for state R3
14	20	Packet type invalid for state P1
15	21	Packet type invalid for state P2
16	22	Packet type invalid for state P3
17	23	Packet type invalid for state P4
18	24	Packet type invalid for state P5
19	25	Packet type invalid for state P6
1A	26	Packet type invalid for state P7
1B	27	Packet type invalid for state D1
1C	28	Packet type invalid for state D2
1D	29	Packet type invalid for state D3
20	32	Packet not allowed
21	33	Unidentifiable packet
22	34	Call on one-way logical channel
23	35	Invalid packet type on a permanent virtual circuit
24	36	Packet on unassigned LCN
25	37	Reject not subscribed to
26	38	Packet too short
27	39	Packet too long
28	40	Invalid GFI
29	41	Restart or registration packet with nonzero in bits 1 to 4 of octet 1 or bits 1 to 8 of octet 2
2A	42	Packet type not compatible with facility
2B	43	Unauthorized interrupt confirmation
2C	44	Unauthorized interrupt
2D	45	Unauthorized reject
30	48	Timer expired
31	49	Timer expired for incoming call
32	50	Timer expired for clear indication
33	51	Timer expired for reset indication

Code (Hex)	Code (Dec)	Description
34	52	Timer expired for restart indication
35	53	Timer expired for call deflection
40	64	Call set up, clearing, or registration problem
41	65	Facility code not allowed
42	66	Facility parameter not allowed
43	67	Invalid called address
44	68	Invalid calling address
45	69	Invalid facility length
46	70	Incoming call barred
47	71	No logical channel available
48	72	Call collision
49	73	Duplicate facility requested
4A	74	Nonzero address length
4B	75	Nonzero facility length
4C	76	Facility not provided when expected
4D	77	Invalid CCITT-specified DTE facility
4E	78	Maximum number of call redirections or deflections exceeded
50	80	Miscellaneous
51	81	Improper cause code for DTE
52	82	Octet not aligned
53	83	Inconsistent Q bit setting
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

Diagnostic codes with values of 80 or greater in hexadecimal, or with values of 128 or greater in decimal, have been defined for a proprietary network. To learn the meanings of these codes, contact the administrator for that network.

Appendix B

Technical Support Information List



When a problem arises that you are unable to resolve, the resource of last resort is your router technical support representative. To analyze a problem, your technical support representative will need certain information about the situation and symptoms. It will speed the problem isolation process if you are able to present this data when you contact your representative.

Gathering Information About Your Internet

Before gathering any specific data, the first thing to do is compile a list of all symptoms that users have reported on the internetwork (such as connections dropping or slow host responsiveness).

The next step is to gather specific information. Typical information needed to troubleshoot internetworking problems falls into two general categories: information required for any situation and information specific to the topology/problem.

Information always needed by technical support engineers includes the following:

1. Configuration listing of all routers involved
2. Complete specifications of all routers involved
3. Version numbers of software (obtained with **show version** command) and firmware (obtained with the **show controllers** command) on all routers
4. Network topology map, including any suspected back doors
5. List of hosts and servers (host and server type, number on network, description of host operating systems implemented)
6. List of network layer protocols, versions, vendors

Specific requirements that vary depending on the situation:

1. Output from general **show** commands:
show interfaces
show controllers {serial | token | mci | cbus | fddi}
show processes

2. Output from protocol-specific **show** commands:
show protocol-type route
show protocol-type traffic
show protocol-type interfaces
show protocol-type arp
show apple global (AppleTalk only)
show novell servers (Novell only)
3. The relevant **debug** diagnostic EXEC commands



Caution: Throughout this publication, the use of **debug** commands is suggested for obtaining information about network traffic and router status. Use these commands with great care. In general, it is recommended that these commands only be used under the direction of your router technical support representative when troubleshooting specific problems. Enabling debugging can disrupt operation of the router when internets are experiencing high load conditions. When you finish using a **debug** command, remember to disable it with its specific **undebug** command or with the **undebug all** command.

4. The output from protocol-specific **ping** (Echo Request/Echo Reply) and **trace** diagnostic tests
5. Network analyzer traces
6. Core dumps (use **exception dump** global configuration command). You can also use the **write core** command with Software Release 9.0 or higher (if the system is operational)

Getting the Data from Your Router

You must tailor the way you obtain information from the router to the systems you are using to get that information. A few hints are outlined in the following list (organized by information-gathering tool).

For PC and Macintosh

Connect PC or Macintosh to the console port of the router and log all output to a disk file. The exact procedure varies depending on the communication package used with the PC.

For Terminal Connected to Console Port or Remote Terminal

The only way to get information with this configuration is to attach a printer to the AUX port on the terminal (if one exists) and force all output to the screen to go to the printer. Using a terminal is undesirable because there is no way to capture the data to a file.

For UNIX Workstation

At your UNIX prompt, enter the command **script** *filename*, then Telnet to the router. The UNIX **script** command causes all screen output to be captured to the filename specified. Enter **exit** to stop capturing and close the file.

Note: To get your system to automatically log specific error messages or operational information to a UNIX syslog server, use the **logging** *internet-address* global configuration command. Refer to your *Router Products Configuration and Reference* publication for more information about using the **logging** command and setting up a syslog server.

Presenting Data to Your Technical Support Representative

Your technical support representative will accept information in any format that you can provide. Common forms include data sent via file transfer, electronic mail, magnetic media, and hard copy. The order of preference is as follows:

- The preferred method of information delivery is for you to deliver information via the File Transfer Protocol (FTP) service over the Internet. If your environment supports FTP, you can place your file in the “incoming” directory on the host named *ftp.cisco.com*.
- The next best method is to send data by electronic mail. Before trying this method, be sure to contact your router technical support representative, especially when transferring binary core dump files.
- Third on the list is transfer via a PC-based communications protocol, such as *Kermit*. Again, be sure to contact your technical support representative before attempting any transfer.
- Fourth on the list is transfer by disk or tape.
- The least favorable method is hard copy transfer by physical mail or fax.

Appendix C

Problem-Solving Checklist/Worksheet



To isolate problems in your internetwork, you must first compile all the relevant facts and then methodically address each suspect problem. This appendix provides a troubleshooting checklist and general worksheet to help you in this process. Use the checklist and worksheet provided here as *initial guidelines* to assist you in developing your own checklist and worksheet—one tailored to your own internetworking environment.

Troubleshooting Checklist

Before you start making any changes to your internet, be sure you can answer the following questions positively:

1. Have you identified and compiled a list of all the reported symptoms on your internet?
2. Do you know your internetwork? Do you have an accurate physical and logical map of your internet?
3. Do you have a list of all the network protocols implemented in your network?
4. Do you know which protocols are being routed?
5. Do you know which protocols are being bridged?
6. Do you know all the points of contact to external networks?
7. For every symptom, have you developed a list of potential problems and causes?
8. For each problem, do you have a plan of action?

If you can answer *yes* to these questions, you can begin the process of problem isolation. Remember: eliminate one problem at a time.

Troubleshooting Worksheet

1. Symptoms reported:

2. Network Topology Map—attach separate sheet(s)

3. Protocols routed:

4. Protocols bridged:

5. Media used in your environment:

6. Internet equipment (including network address, vendor, model, and function):

7. Suspect end system and internet nodes (including network address, vendor, model, and function):

8. Applications being used on the network (FTP, sendmail, NFS, NetWare, etc.):

9. Symptoms and likely problems:

[illegible]

10. Action plan for each problem:

[illegible]

11. Action outcomes:

[illegible]

Appendix D

Creating Core Dumps



Caution: Use the commands discussed in this appendix only in coordination with a qualified technical support representative. The resulting binary file must be directed to a specific syslog server and subsequently interpreted by qualified technical personnel.

When the router crashes, it can be very useful to obtain a full copy of the memory image (core dump) to analyze the cause of the crash. Core dumps generally are only useful to your router technical support representative.

Note: To obtain a core dump, the router must be running Software Release 8.3 or later.

To obtain a core dump, use the **exception dump** *IP-address* global configuration command. *IP-address* is the address of your TFTP server. The core dump is written to a file named *hostname-core* on your TFTP server, where *hostname* is the name of the router, as assigned using the **hostname** global configuration command. Using this command causes the router to attempt to make a core dump when it crashes.

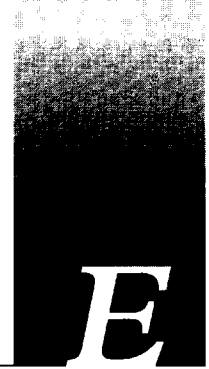
This procedure cannot be guaranteed to work. It can fail if the system crash is serious. If successful, the core dump file will be the size of the memory available on the processor (for example, 4 Mbytes for a CSC/3).

Depending on your TFTP server, you may need to create a target file before the router can write to it. You can test this by attempting to use the TFTP **put** command from a workstation.

You also can test core dumps by using the EXEC command **write core**. This command causes the router to generate a dump and is useful if the router is malfunctioning but has not crashed.

Appendix E

References and Recommended Reading



This appendix lists technical publications—many of which are available commercially—that you may find useful when troubleshooting internetworks.

Commercially Available Publications

Held, Gilbert. *Data Communications Testing and Troubleshooting, Second Edition*. Van Nostrand Reinhold, 1992.

Jones, Nancy E.H., and Kosiur, Dave. *Macworld Networking Handbook*. IDG Books Worldwide, Inc., 1992.

Malamud, Carl. *Analyzing DECnet/OSI Phase V*. Van Nostrand Reinhold, 1991.

Malamud, Carl. *Analyzing Novell Networks*. Van Nostrand Reinhold, 1990.

Malamud, Carl. *Analyzing Sun Networks*. Van Nostrand Reinhold, 1992.

Miller, Mark A. *LAN Protocol Handbook*. M&T Publishing, 1990.

Miller, Mark A. *LAN Troubleshooting Handbook*. M&T Publishing, 1989.

Miller, Mark A. *Troubleshooting Internetworks*. M&T Publishing, 1991.

Perlman, Radia. *Interconnections: Bridges and Routers*. Addison-Wesley Publishing Company, Inc. 1992.

Technical Publications and Standards

IBM. *Token-Ring Problem Determination Guide*. SX27-3710-04, 1990.

Apple Computers. *Inside AppleTalk*. Addison-Wesley Publishing Company, Inc. 1991.

Apple Computers. *Planning and Managing AppleTalk Networks*. Addison-Wesley Publishing Company, Inc. 1991.



Index

A

access lists

AppleTalk

problems, 3-6

Novell IPX, 5-6, 5-9

TCP/IP

extended, 2-48

standard, 2-47

Address Resolution Protocol

See ARP

AppleTalk

adding zone names, 3-8

backdoor routes, 3-7

common problems, 3-3

connectivity scenario, 2-3

diagnostic tips, 3-3

discovery mode

disabling, 3-25

enabling, 2-11, 3-13, 3-14, 3-26

duplicate network numbers, 2-7

extended and nonextended networks, 3-2

ghost zones, 3-27

NBP, 3-5

nonseed router, 3-13, 3-14, 3-16, 3-18, 3-27

Phase 1 and Phase 2, 3-2

Phase 1/Phase 2 rule violations

finding, 2-8

port stuck in acquiring mode, 3-25

problem prevention tips, 3-7

problems

backdoor routes, 3-7, 9-4

bad access lists, 3-6, 3-15, 3-19, 3-22

configuration mismatch, 3-4, 3-13, 3-14,
3-16, 3-18, 3-26

conflicting zone lists, 3-23

crossed serial circuits, 3-25

duplicate network numbers, 2-7, 3-5, 3-15,
3-20, 3-22, 9-3

ghost zones, 3-27

network congestion, 3-21

old network numbers not removed, 3-27

Phase 1/Phase 2 rule violations, 2-9, 3-5,
3-15, 3-17

Phase 1-only routers, 2-10

router configured for discovery mode, 3-25

unstable routes, 3-7, 3-19, 3-21, 3-23, 3-24

ZIP storms, 2-7, 3-19, 3-20, 3-22, 9-3

seed router, 3-25

terminology, 3-1

ZIP storms

finding, 2-7

See also diagnosing, symptoms, troubleshooting

appletalk address command

assigning network numbers, 3-25

enabling discovery mode, 2-11, 3-13, 3-14, 3-26

appletalk cable-range command, 3-25

appletalk discovery command

disabling discovery mode, 3-25

appletalk event-logging command, 3-7

appletalk name-lookup-interval command

enabling NBP name registration, 3-13, 3-14

used with ping feature, 3-4, 3-11

appletalk proxy-nbp command, 2-10, 3-15

appletalk timers command

adjusting, 3-19

reducing congestion problems, 3-21

resolving unstable routes, 3-8, 3-21

ARP

comparing ARP and RIF tables, 2-18

B

backdoor bridge

AppleTalk, 3-7

Novell IPX, 2-41, 5-6

backdoor route

See backdoor bridge

bandwidth command, 8-23

beaconing, 4-5

bridge-group lat-compression command, 7-24, 9-13

bridging

- IBM internet problems, 2-21
- LAT problems, 7-24, 8-27
- Novell IPX problems, 8-5
- remote SRB problems, 4-9
- SRB problems, 4-4, 4-5, 4-6, 4-8
- SRT problems, 4-12
- SRT/SRB incompatibilities, 2-24
- translation problems, 4-10
- translation scenario, 2-21

buffers

- hardware, 7-20
- internal, message logging, 10-5
- system, 7-20

buffers command

- determining number to use, 7-21
- reducing performance problems, 7-20

C

cause codes

- X.25, A-1

CiscoWorks

- using to troubleshoot problems, 1-17

clear arp command, 2-18, 2-57

clear counters command

- using to troubleshoot serial lines, 8-24

clear rif command, 2-18

client problems

- See host problems

configuration examples

- AppleTalk (complete), 2-13
- HDLC (complete), 8-26
- IBM, 2-19, 2-29
- Novell IPX, 2-43
- TCP/IP (complete), 2-50
- TCP/IP (priority queuing), 8-20
- X.25 (complete), 2-60

configuration problems

- See “problems” for specific protocols and technologies

connectivity problems

- See “problems” for specific protocols and technologies

console

- logging messages to, 10-4

core dumps

- obtaining, D-1
- using, 1-9

CSU/DSU

- local and remote loopback tests, 7-15

D

debug ? command, 10-2

debug all command, 10-2

debug apple-arp command, 10-9

debug apple-errors command, 10-10

debug apple-events command, 3-7, 3-19, 3-21, 10-12

debug apple-nbp command, 10-16

debug apple-packet command, 10-19

debug apple-routing command, 10-21

debug apple-zip command

- description, 10-23

- finding ZIP storms, 3-6

debug arp command

- description, 10-24

- SMDS troubleshooting, 7-35

- suggested use, 7-14

debug broadcast command, 10-25

debug command

- disabling, 1-9

- troubleshooting serial lines, 7-13

- using, 1-9

debug commands

- general, 10-1

debug decnet-connects command, 10-27

debug frame-relay command, 10-28

debug frame-relay-events command, 7-14, 7-34, 10-30

debug frame-relay-lmi command, 7-14, 7-33, 10-31

debug frame-relay-packets command, 10-34

debug ip-icmp command, 6-8, 10-36

debug ip-igrp command, 6-6, 10-40

debug ip-igrp-events command, 10-42

debug ip-ospf-events command, 10-43

debug ip-packet command, 10-44

debug ip-rip command, 6-6, 10-47

debug ip-tcp command, 10-48

debug lapb command, 7-14, 10-50

debug lnm-events command, 10-54

debug lnm-llc command, 10-56

- debug lnm-mac command, 10-59
- debug local-ack-state command, 10-61
- debug novell-packet command, 2-35, 2-38, 5-11, 10-62
- debug novell-routing command, 10-63
- debug novell-sap command, 5-9, 10-64
- debug packet command, 10-68
- debug rif command, 10-70
- debug serial-interface command
 - description, 10-74
 - for an MK5025 device, 10-80
 - for DDR, 10-75
 - for frame relay, 10-75
 - for HDLC, 10-76
 - for HSSI, 10-78
 - for ISDN Basic Rate, 10-79
 - for PPP, 10-81
 - for SMDS, 10-82
 - keepalive counters, 7-15, 7-30
 - SMDS troubleshooting, 7-35
 - suggested use, 7-14
- debug serial-packet command
 - description, 10-83
 - for DDR, 10-83
 - for PPP, 10-83
 - for SMDS, 10-85
 - SMDS troubleshooting, 7-35
 - suggested use, 7-14
- debug source-bridge command, 10-86
- debug source-event command, 10-86
- debug span command
 - DEC spanning tree example, 10-92
 - description, 10-91
 - IEEE spanning tree example, 10-91
- debug stun-packet command, 4-20, 4-21
- debug tftp command, 10-94
- debug token-ring command, 10-95
- debug vines-arp command, 10-98
- debug vines-echo command, 10-99
- debug vines-packet command, 10-100
- debug vines-routing command, 10-101
- debug vines-table command, 10-102
- debug x25 command, 10-105
- debug x25-events command, 2-58, 7-14, 7-26, 10-109
- debug x25-vc command, 10-110
- debug xns-packet command, 10-103
- debug xns-routing command, 10-104

DEC LAT

- compression, 7-24, 9-13
- sensitivity to delays, 7-23
- sensitivity to dropped packets, 8-25, 9-13
- translational bridging problems, 4-10

DECnet

- sensitivity to dropped packets, 7-20, 7-22, 8-24

default-metric command, 2-47, 6-15

diagnosing

AppleTalk

- access list problems, 3-15
- common techniques, 3-10
- congestion problems, 3-21
- connections to services drop, 3-24
- connectivity, 2-5
- duplicate network addresses, 2-7
- interface not initializing, 3-16
- intermittent service availability, 3-20
- missing zones, 3-18
- network not visible, 3-14
- old zone names appear, 3-26
- Phase 1/Phase 2 rule violations, 2-9
- poor performance, 9-3
- port stuck in acquiring mode, 3-25
- services cannot be accessed, 3-22
- sporadic service availability, 9-3
- unstable zone lists, 3-23
- ZIP storms, 2-7
- zones and services missing, 3-13

CiscoWorks tools, 1-17

Ethernet

- general problems, 1-20
- verifying connectivity, 2-56

FDDI

- general problems, 1-21

Frame Relay

- new router connectivity, 7-33

general

- system tools, 1-8

hardware

- general, 1-10
- physical inspection, 1-10
- problems at power-up, 1-11
- testing and verifying operation, 1-13

diagnosing (continued)

IBM

- blocked connection over SDLLC, 4-13
- blocked routing over SRB, 4-4
- blocked SRB traffic, 4-6
- blocked traffic over remote SRB, 4-8
- blocked traffic over SRT, 4-12
- blocked traffic over translational bridge, 4-10
- end system problems, 2-17
- ES-to-IS incompatibilities, 2-23
- failed SDLC sessions, 4-20
- intermittent connectivity over SDLC transport, 4-16
- intermittent failures over SRB, 4-9
- LNLM problems, 4-23
- mixed internet problems, 2-23
- NetBIOS connectivity problems, 4-22
- physical layer problems, 4-19
- slow performance over remote SRB, 9-5
- SRB connectivity, 2-16
- SRB/SRT incompatibilities, 2-24
- unable to connect to Token Ring, 4-17
- unexpected failure of SRB network, 4-5

Novell IPX

- backdoor bridge, 2-41
- blocked NetBIOS traffic, 5-11
- blocked SAP updates, 5-9
- blocked traffic over packet-switched network, 5-21
- connectivity, 2-34
- duplicate MAC addresses, 2-39
- duplicate network numbers, 2-37
- encapsulation mismatches, 2-38
- general, 5-1
- helper address problems, 2-40
- interface status, 2-38
- missing SAP server updates, 2-36
- no communication with NetWare servers, 5-4
- physical connections, 2-35
- poor performance after bandwidth upgrade, 8-4
- poor performance after switch to routing, 8-6
- poor performance between rings, 8-8

poor performance over Ethernet

- backbone, 8-10

poor server performance in LAN, 9-6

poor server performance in WAN, 9-7

slow performance over matching parallel links, 8-14

slow performance over unequal parallel links, 8-16

serial lines

basic diagnostic fields, 7-6

extended ping tests, 7-16

general problems, 1-20

line status, 7-3

local loopback tests, 7-15

new router connectivity, 7-30

remote loopback tests, 7-16

using show interfaces command, 7-2

SMDS

new router connectivity, 7-35

TCP/IP

blocked access to certain hosts, 6-7

blocked access to certain services, 6-8

blocked access to offnet hosts, 6-4

blocked access to parts of network, 6-13

blocked access to some networks, 6-6

blocked traffic between domains, 6-15

blocked traffic through backup path, 6-9

connectivity, 2-46

duplicate routing updates appear, 6-11

general, 6-1

performance, 8-18

problem isolation process, 8-18

route redistribution problems, 2-47

routing fails for certain protocols, 6-12

slow performance, 9-8

slow performance over parallel links, 9-9

Token Ring

general problems, 1-20

WAN

bad virtual circuit sequencing, 2-59

bad X.25 configuration, 2-57

clocking problems, 7-18

connections die under load, 7-27, 7-28

connections die unpredictably, 7-29

diagnosing (continued)
 intermittent connectivity, 7-26
 loss of connection, 9-13
 performance, 8-22
 physical media, 2-53
 problem isolation process, 8-22
 selectively blocked connectivity, 7-37
 serial interface, 2-53
 serial lines, 7-1
 slow host or network response, 9-11
 verifying host communication, 2-56
 X.25 connectivity, 2-52

X.25

 new router connectivity, 7-31

diagnostic codes

 X.25, A-3

dial backup, 7-28, 9-11, 9-13

distance command, 6-15, 9-10

document conventions, xxix

duplicate network numbers

 AppleTalk, 2-7

 Novell IPX, 2-37, 5-6

E

encapsulation

 Novell IPX support, 2-39

end station problems

 See host problems

error messages

 IBM

 open lobe fault, 4-17

 ICMP, 10-45

 setting levels, 10-4

Ethernet

 maximum packet size, 4-12

 Novell IPX

 frame types, 5-8

 problems

 mapping to Token Ring addresses, 4-10

 noise, 1-20

 translational bridge problems, 4-10

 See also diagnosing, symptoms

ethernet-transit-oui command, 2-26, 4-11

exception dump command

 description, D-1

 using core dumps, 1-9

F

fast switching

 disabling, 7-21, 8-24, 9-11, 9-13

FDDI

 not functioning, 1-21

 See also diagnosing, troubleshooting

flapping, 3-7

Frame Relay

 problems

 bad access list, 7-34

 bad cabling, 7-34

 dead hardware, 7-34

 misconfigured dynamic mapping, 7-33

 misconfigured static map, 7-34

 misconfigured switch, 7-33

 wrong keepalive setting, 7-33

 See also diagnosing, symptoms

frame-relay encapsulation command, 5-21

frame-relay map novell command, 5-21

G

ghost zones, 3-27

H

hardware

 See troubleshooting, symptoms

helper addresses

 Novell IPX

 all nets broadcast, 5-15

 basic configuration, 5-13

 behavior over parallel links, 5-18

 directed broadcast, 2-42, 5-16

 multiple helper addresses per interface, 5-17

 multiple serial lines, 5-15

 NetBIOS server broadcast, 5-17

 serial line configuration, 5-14

hold-queue command, 7-22

host problems

AppleTalk

- Macintosh broadcasts, 3-8
- server is Phase 1 only, 2-10

IBM

- end station does not support RIF, 2-24, 4-6
- frame size mismatch, 4-12
- possible SRB software bug, 2-18, 4-5
- XID mismatch, 4-13

Novell IPX

- clients not attached to network, 2-35, 5-4
- limited-user version of NetWare, 2-38, 5-10
- misconfigured network number, 5-21
- server cannot handle SAP update frequency, 5-10
- server not sending SAP updates, 5-9
- servers not attached to network, 2-36, 5-4

TCP/IP

- back doors through UNIX hosts, 2-46
- misconfigured subnet mask, 6-4, 6-7, 6-13
- no default gateway, 6-4, 6-6, 6-7, 6-13

WAN

- host not sending ARPs, 7-37
- host points at wrong router, 7-37
- using ping to verify reachability, 2-56

See also "problems" for specific protocols and technologies

hostname command

- using with exception dump command, D-1

I

IBM

- 3174 cluster controller, 2-21
- 3270, 2-22
- 3745 FEP, 2-21
- AS/400, 2-21
- connectivity scenario, 2-15, 2-21
- general diagnostics, 4-2
- LAT translation problems, 4-10
- LLC2 timers, 4-9, 4-16
- locally administered addresses, 4-15
- problems
 - bad ring speed specification, 4-18

bridging protocols that require

- routing, 4-11, 4-12

broken equipment, 4-21

broken SDLC physical connections, 4-20

end station does not support RIF, 4-6, 4-8

end station sending spanning explorers, 4-7

ES-to-IS incompatibilities, 2-23

hardware does not support SRT, 4-12

high CPU utilization, 9-5

hop count exceeded, 4-8

host communication, 2-17

incorrect NetBIOS name cache

- mapping, 4-22

incorrect source-bridge remote-peer

- specification, 4-22

loops in translational bridge

- environment, 4-11

misconfigured IP addresses, 2-17

misconfigured LNM MAC address, 4-23

misconfigured ring number, 4-6

misconfigured source-bridge

- commands, 4-8

misconfigured stun peer command, 4-20

misconfigured stun route address

- command, 4-20, 4-21

missing multiring command, 2-16, 2-26, 4-4

missing partner command, 2-27, 4-13

missing sdlc xid command, 2-27, 4-13

missing SRB drivers in end systems, 2-17

no route to remote peer, 4-8

packets too large for Ethernet, 4-12

physical layer mismatch, 4-19

relay open at MAU, 4-17

ring beaconing, 4-5

router does not support media address

- mapping, 4-10

SDLC clock settings, 4-19

SDLC timing, 4-16

SDLLC microcode incompatibility, 4-13

SDLLC serial signal mismatch, 4-13

SDLLC V.35 jumper setting, 4-13

secondary link physical connectivity, 4-21

- serial link problems, 4-8
- sessions timing out, 4-9
- SRT dropping packets with RIF, 4-12
- SRT/SRB incompatibilities, 2-24
- vendor code mismatch, 2-26, 4-11
- remote SRB, 4-8, 4-9
- router as DTE, 4-19
- RS-232 signal requirements, 4-14
- SDLLC, 4-13
- SRB, 4-4, 4-5, 4-6
- SRT, 4-12
- translational bridging, 4-10
- See also diagnosing, symptoms, troubleshooting
- IBM SDLC transport, 4-16
- ICMP
 - error messages, 10-45
- IGRP
 - route redistribution problems, 6-15
- Interior Gateway Routing Protocol
 - See IGRP
- internal buffer
 - message logging, 10-5
- ip access-group command, 2-48, 6-6
- ip route-cache command, 7-21
- IPSO
 - security actions table, 10-45

L

- LAN Network Manager
 - See LNM, 1-21
- LAT
 - See DEC LAT
- LLAP
 - routers, 3-8
- LNM
 - problems linking router, 4-23
- Local Area Transport
 - See DEC LAT
- LocalTalk Link Access Protocol
 - See LLAP
- logging buffered command, 10-5
- logging command, 10-6
- logging console command, 10-4
- logging monitor command, 10-5
- logging trap command, 10-6

M

- MAC address
 - Novell IPX
 - mapping to Frame Relay DLCI, 5-21
 - mapping to X.25, 5-21
 - nonunique address, 5-10
 - problems with translational bridging, 4-10, 4-12
- mac-address command, 4-23
- media
 - problems
 - general, 1-20
- Media Access Control
 - See MAC address
- message logging
 - keywords and levels, 10-4
 - to a UNIX syslog server, 10-6
 - to another monitor, 10-5
 - to internal buffer, 10-5
 - to the console, 10-4
- messages
 - ICMP, 10-45
- monitor
 - logging messages to, 10-5
- multiring command, 2-16, 2-26

N

- Name Binding Protocol
 - See NBP
- NBP
 - Phase 1/Phase 2 differences, 3-5
- netbios name-cache command, 4-22
- NetWare
 - See Novell NetWare
- network analyzer
 - IBM
 - examining the RIF, 2-24
 - looking for nonzero high order bit, 2-23
 - Novell IPX
 - looking for routing and SAP updates, 5-6
- nonseed router, 3-13, 3-14, 3-16, 3-18, 3-27
- novell access-group command, 5-6, 5-12
- novell encapsulation arpa command, 5-8
- novell helper-address command, 2-40, 5-11
- novell input-sap-filter command, 5-9

Novell IPX

- access lists, 5-6, 5-9
- backdoor bridge, 2-41, 5-6
- BNETX.COM client software, 8-6, 8-8
- configuring helper addresses, 2-40
- connectivity scenario, 2-33
- directed broadcasts, 2-40
- Ethernet encapsulation, 2-39, 5-8
- flooding broadcasts, 2-40
- helper addresses
 - alternatives, 5-13
 - problems if missing, 5-11
- LIPX.NLM module, 8-6, 8-8
- mapping to Frame Relay addresses, 5-23
- mapping to X.25 addresses, 5-22
- NetBIOS, 2-33
- NetBIOS broadcast requirements, 5-11
- NetWare 2.15 and 2.2, 5-3
- NetWare 2.15 network numbers, 5-5
- NetWare 3.11 network numbers, 5-5
- PBURST.NLM module, 8-6, 8-8
- problems
 - backdoor bridge, 2-41, 5-6
 - bad access lists, 5-6, 5-9, 5-12
 - congestion, 8-10
 - dead interface, 5-4
 - duplicate MAC addresses, 2-39
 - duplicate network numbers, 2-37, 5-6, 5-10
 - encapsulation errors, 5-21
 - encapsulation mismatches, 2-38
 - excessive traffic, 9-6
 - Frame Relay mapping error, 5-21
 - helper address configuration, 2-40
 - insufficient bandwidth, 8-4, 9-6
 - isolating, 5-2
 - limited user version, 2-38
 - limited-user server software, 5-10
 - maximum packet size limitation, 8-6
 - misconfigured network numbers, 5-5, 5-10, 5-21
 - mismatched encapsulation methods, 5-8
 - missing or misconfigured novell helper-address command, 5-11
 - network number mismatch, 5-5
 - nonfunctional Ethernet, 5-7

- nonfunctional FDDI, 5-7
- nonfunctional serial lines, 5-7
- nonfunctional Token Ring, 5-8
- nonunique MAC address, 5-10
- other protocol dominating CPU, 9-7
- physical attachment to network, 2-35
- ring speed mismatch, 5-9
- RIP, 5-5, 5-10
- router not load balancing, 8-14, 8-16
- SAP updates not being sent, 2-36, 5-9
- server not sending SAP updates, 5-9, 5-10
- servers cannot handle router SAP update rate, 5-10
- servers not attached to network, 5-4
- software support of LIPX.NLM, 8-8
- X.25 mapping error, 5-21
- sensitivity to dropped packets, 7-22
- translation of encapsulation types, 2-39
- See also diagnosing, symptoms, troubleshooting

novell maximum-paths command, 8-10

Novell NetWare

- 2.15 network numbers, 5-5
- 3.11 network numbers, 5-5
- clients, 5-4
- display servers command, 5-9
- load monitor command, 9-6
- servers, 5-4
- slist command, 5-9, 5-10
- track on command, 2-35

novell network command, 2-37

novell output-sap-delay command, 5-10

novell output-sap-filter command, 5-9

novell routing command, 2-36, 5-5

O

- offset-list command, 9-10
- open lobe fault message, 4-17

P

- partner command, 2-27, 4-13
- passive command, 2-47
- performance
 - AppleTalk
 - common problems, 9-3

- IBM
 - common problems, 9-5
- Novell IPX
 - 16-Mbps Token Ring scenario, 8-7
 - bandwidth upgrade scenario, 8-3
 - common problems, 9-6, 9-7
 - Ethernet backbone scenario, 8-9
 - matching parallel links scenario, 8-13
 - switch to routing scenario, 8-5
 - unequal parallel links scenario, 8-15
- scenario summaries, 8-2
- TCP/IP
 - common problems, 9-8, 9-9
 - serial internet scenario, 8-17
- WAN
 - common problems, 9-11, 9-13
 - slow HDLC link scenario, 8-21
- ping command
 - AppleTalk
 - finding problem nodes, 3-11
 - general, 1-9
 - IBM
 - isolating problems in SRB, 2-18
 - TCP/IP
 - isolating problems, 2-47, 6-2, 6-6, 8-18
 - troubleshooting Frame Relay, 7-33
 - troubleshooting serial lines, 7-16
 - WAN
 - isolating serial problems, 2-56
- preventive measures
 - AppleTalk, 3-7
 - IBM SDLLC, 4-15
- priority queuing
 - enabling, 8-27, 9-11, 9-13
 - Novell IPX, 9-7
 - when to use, 7-23
- priority-list command, 7-23, 8-27
- problems
 - See "problems" for specific protocols and technologies
- problem-solving model
 - defined, 1-3
- protocol analyzer
 - See network analyzer

R

- reachability problems
 - See "problems" for specific protocols and technologies
- references
 - commercially available publications, E-1
 - technical publications and standards, E-1
- RIF
 - examining for XID to NULLSAP, 2-18
 - using network analyzer to examine, 2-24
- ring speed command, 4-18, 5-8, 5-9
- RIP
 - Novell IPX
 - network number problems, 5-5
 - route redistribution problems, 6-15
- route flapping
 - See flapping
- route redistribution
 - TCP/IP, 2-47
- Routing Information Field
 - See RIF
- Routing Information Protocol
 - See RIP

S

- SAP
 - updates not being sent by server, 2-36, 5-9
- scenarios
 - AppleTalk connectivity, 2-3
 - connectivity summary, 2-2
 - defined, 1-6
 - IBM connectivity, 2-15, 2-21
 - Novell IPX connectivity, 2-33
 - Novell IPX performance
 - bandwidth upgrade, 8-3
 - interconnected rings, 8-7
 - matching parallel links, 8-13
 - problems after switch to routing, 8-5
 - slow performance over Ethernet
 - backbone, 8-9
 - slow performance over unequal parallel links, 8-15
 - performance summary, 8-2
 - SDLC connectivity, 2-21
 - SDLLC connectivity, 2-21

- SRB connectivity, 2-15
- SRT connectivity, 2-21
- TCP/IP connectivity, 2-45
- TCP/IP performance, 8-17
- translational bridging connectivity, 2-21
- WAN performance, 8-21
- X.25 connectivity, 2-51
- SDLC
 - blocked connectivity, 4-16
 - connectivity scenario, 2-21
- SDLLC
 - blocked connectivity, 4-13
 - connectivity scenario, 2-21
 - virtual ring address considerations, 4-15
- sdllc traddr command, 4-15
- sdllc xid command, 2-27, 4-13
- security
 - ICMP error messages, 10-45
 - using access lists, 2-47
- seed router, 3-25
- serial lines
 - basic diagnostic information, 7-6
 - IBM RS-232 signals, 4-14
 - local loopback tests, 2-55
 - problems
 - congestion, 7-26, 7-27, 9-13
 - dead hardware, 7-29
 - EMI, 7-28
 - keepalives not being received, 7-30
 - link is dead, 7-30
 - noise, 7-27, 9-11, 9-13
 - not functioning, 1-20
 - overutilized bandwidth, 7-20
 - unreliable hardware, 9-12, 9-13
 - See also diagnosing, symptoms
- server problems
 - See host problems
- Service Advertisement Protocol
 - See SAP
- show access-lists command, 6-7, 6-8
- show appletalk arp command, 3-13
- show appletalk global command, 2-9, 3-15
- show appletalk interface command, 2-7, 3-4, 3-13, 3-25
- show appletalk neighbor command, 2-9, 2-10, 3-15
- show appletalk route command, 2-7, 2-8, 3-6, 3-19, 9-3
- show appletalk traffic command, 2-7, 2-11, 3-6, 3-19, 9-3
- show appletalk zones commands, 3-27
- show arp command
 - SMDS troubleshooting, 7-35
 - used to determine hardware address, 2-18
- show buffers command
 - using to troubleshoot serial lines, 7-21, 8-24
- show command
 - creating a topology map, 1-8
 - using, 1-8
- show configuration command, 3-10
- show controllers command, 2-39, 2-53
- show controllers mci command, 2-53, 7-13
- show frame-relay map command, 7-33, 7-34
- show interface serial command
 - diagnosing SDLC problems, 4-21
- show interfaces command, 8-18, 9-4, 9-9
- show interfaces ethernet command, 2-56, 5-7
- show interfaces fddi command, 1-21, 5-7
- show interfaces serial command
 - abort field, 7-8
 - carrier transitions, 7-10
 - CRC field, 7-7
 - determining operational status, 2-54
 - evaluating input errors, 7-6
 - frame field, 7-8
 - input drops, 7-9
 - inspecting interface status, 2-55
 - interface resets, 7-9
 - output drops, 7-8
 - using to troubleshoot, 7-2
 - using to verify congestion problems, 8-23
 - X.25-specific fields, 7-11
- show interfaces token command, 4-18, 5-8
- show ip arp command, 6-2
- show ip interface command, 6-7, 6-8
- show ip protocol command, 6-11
- show ip route command, 6-2, 6-6
- show ip traffic command, 9-9
- show lnm config command, 4-23
- show logging command, 10-5, 10-6
- show novell interface command, 5-5
- show novell servers command, 2-36, 5-6, 5-10
- show novell traffic command, 2-38

-
- show process command, 9-5, 9-7
 - show rif command, 2-18, 4-22
 - show stun command, 4-20
 - show version command, 2-53
 - SMDS
 - problems
 - bad access list, 7-36
 - bad cabling, 7-36
 - dead hardware, 7-36
 - misconfigured multicast address, 7-35
 - misconfigured router, 7-35
 - misconfigured static mapping, 7-36
 - misconfigured switch, 7-35
 - See also diagnosing, symptoms
 - smds address command, 7-35
 - smds enable-arp command, 7-36
 - smds encapsulation command, 7-35
 - smds multicast command, 7-35
 - Sniffer
 - trace
 - examining a RIF, 2-25
 - high-order bit, 2-24
 - See also network analyzer
 - Source Route Bridge
 - See SRB
 - Source Route Transparent
 - See SRT
 - source-bridge fst-peername command, 9-5
 - source-bridge remote-peer command, 4-8, 4-9, 4-22, 9-5
 - source-bridge ring-group command, 4-15
 - source-bridge spanning command, 4-7
 - SRB
 - blocked connectivity, 4-6
 - connectivity fails unexpectedly, 4-5
 - connectivity scenario, 2-15
 - detecting incompatibilities with SRT, 2-24
 - NetBIOS problems, 4-22
 - routing blocked over SRB, 4-4
 - See also IBM
 - SRT
 - blocked connectivity, 4-12
 - connectivity scenario, 2-21
 - detecting incompatibilities with SRB, 2-24
 - stun peer-name command, 4-20
 - stun route address command, 4-16, 4-20
 - subnet masks, 6-14
 - symptom modules
 - defined, 1-5
 - symptoms
 - AppleTalk
 - connections to services drop, 3-24
 - connectivity scenario, 2-3
 - intermittent service availability, 3-20
 - missing zones, 3-18
 - network not visible, 3-14
 - old zone names appear, 3-26
 - poor performance, 9-3
 - port stuck in acquiring mode, 3-25
 - router interface inactive, 3-16
 - sporadic service availability, 9-3
 - summary list, 3-12
 - unstable zone lists, 3-23
 - users cannot access visible services, 3-22
 - zones and services not visible, 3-13
 - Frame Relay
 - cannot connect over new router, 7-33
 - hardware
 - breaker trips or fuse blows, 1-12
 - card and chassis failures, 1-14
 - cards not detected at power-up, 1-12
 - constant or partial reboot, 1-12
 - no blower, 1-12
 - no fan, 1-12
 - no LEDs at boot, 1-12
 - no response from chassis, 1-12
 - system will not boot, 1-12
 - IBM
 - cannot connect over SDLLC, 4-13
 - connectivity scenario, 2-15
 - intermittent connectivity over SDLC, 4-16
 - intermittent failures over SRB, 4-9
 - NetBIOS client cannot connect to
 - servers, 4-22
 - no communication over remote SRB, 4-8
 - no communication over SRB, 4-6
 - no communication over translational
 - bridge, 4-10
 - router cannot attach to Token Ring, 4-17
 - router cannot be linked from LNM, 4-23

symptoms (continued)

- router cannot communicate to SDLC device, 4-19
- routing blocked in SRB network, 4-4
- SDLC sessions not initializing, 4-20
- slow performance over remote SRB, 9-5
- SRB network fails unexpectedly, 4-5
- summary list, 4-3
- traffic blocked through SRT, 4-12

Novell IPX

- blocked SAP updates, 5-9
- cannot communicate with NetWare servers, 5-4
- cannot connect over packet-switched network, 5-21
- connectivity scenario, 2-33
- NetBIOS traffic is blocked, 5-11
- performance scenario, 8-9
- poor performance after serial upgrade, 8-3
- poor performance after switch to routing, 8-5
- poor performance between rings, 8-7
- poor server performance in LAN, 9-6
- poor server performance in WAN, 9-7
- slow performance over parallel links, 8-13, 8-15

performance

- summary list, 9-2

serial lines

- cannot connect over new router, 7-30

SMDS

- new router connectivity, 7-35

TCP/IP

- backup/parallel path not functioning, 6-9
- cannot access certain hosts, 6-7
- cannot access certain networks, 6-6
- cannot access offnet hosts, 6-4
- cannot reach parts of own network, 6-13
- connectivity scenario, 2-45
- duplicate routing updates, 6-11
- performance scenario, 8-17
- routing does not work for certain protocols, 6-12
- slow performance, 9-8
- some services not available, 6-8

- summary list, 6-3

- traffic cannot get from one domain to another, 6-15

WAN

- connections die at a specific time, 7-28
- connections die unpredictably, 7-29
- connections die with load, 7-27
- connections drop, 9-13
- intermittent connectivity, 7-26
- performance scenario, 8-21
- slow host or network response, 9-11
- some users cannot connect, 7-37
- summary list, 7-25
- X.25 connectivity scenario, 2-51

X.25

- cannot connect over new router, 7-31

Synchronous Data Link Control

See SDLC

syslog daemon, 10-7

syslog messages

- levels of, 10-6

syslog server

- limiting messages to, 10-6

syslog server, UNIX

- logging messages to, 10-6

T

TCP/IP

- default gateway specification, 6-4

- load balancing problems, 9-10

problems

- backdoor route, 6-11
- bad access lists, 6-6, 6-7, 6-10, 6-12, 6-13, 9-8, 9-9
- bad administrative distance, 6-15
- bad extended access lists, 6-8
- congestion, 8-19, 9-8
- discontinuous network addressing, 6-6, 6-10
- Ethernet errors, 6-10
- misconfigured load balancing, 9-9
- misconfigured subnet mask, 6-4, 6-7, 6-13
- missing default-metric command, 6-15
- no default gateway, 6-4, 6-6, 6-7, 6-13
- route redistribution, 2-47

- router down between hosts, 6-5
- routing has not converged, 6-10
- serial line errors, 6-10
- slow performance over parallel links, 9-9
- uncontrolled back doors, 2-46
- unreliable hardware, 9-9
- unreliable network link, 9-8
- route redistribution problems, 6-15
- subnet mask errors, 6-14
- See also diagnosing, symptoms, troubleshooting
- terminal monitor command, 10-5
- test equipment
 - general description, 1-19
- Token Ring
 - general diagnostics, 4-2
 - problems
 - mapping to Ethernet addresses, 4-10
 - not functioning, 1-20
 - relay open at MAU, 4-17
 - ring speed modifications, 4-18, 5-8
 - router cannot connect to ring, 4-17
 - translational bridge problems, 4-10
 - vendor code mismatch problem, 2-26
 - See also diagnosing, symptoms
- trace command
 - general, 1-9
 - TCP/IP
 - isolating problems, 6-2, 6-6, 6-8
- translational bridging
 - connectivity scenario, 2-21
 - DEC LAT problems, 4-10
 - isolating problems, 4-10
 - using in place of SRT, 4-12
- troubleshooting
 - AppleTalk
 - connectivity scenario, 2-3
 - connectivity symptoms, 3-1
 - performance symptoms, 9-3
 - checklist, C-1
 - CiscoWorks tools, 1-17
 - connectivity scenarios, 2-1
 - Ethernet, 1-20
 - FDDI, 1-21
 - gathering data for technical support staff, B-1
 - getting data from the router, B-2

- hardware
 - 3000, 1-16
 - 4000, 1-16
 - 500-CS, 1-16
 - ciscoBus backplane, 1-15
 - CSC- C2FCIT, 1-14
 - CSC/2, 1-14
 - CSC/3, 1-14
 - CSC/4, 1-14
 - CSC-1R, 1-15
 - CSC-2R, 1-15
 - CSC-C2FCI, 1-14
 - CSC-CCTL, 1-14
 - CSC-CCTL2, 1-14
 - CSC-CTR, 1-15
 - CSC-ENVM, 1-14
 - CSC-FCI, 1-14
 - CSC-M, 1-15
 - CSC-MC, 1-15
 - CSC-MC+, 1-15
 - CSC-MEC, 1-15
 - CSC-MT, 1-15
 - CSC-R, 1-15
 - CSC-R16M, 1-15
 - FDDI appliques, 1-14
 - IGS, 1-16
 - MCI, 1-15
 - Multibus backplane, 1-15
 - SCI, 1-15
 - serial appliques, 1-15

- IBM
 - connectivity scenario, 2-21
 - connectivity symptoms, 4-1
 - performance symptoms, 9-5
 - SRB connectivity scenario, 2-15
- Novell IPX
 - bandwidth upgrade problem, 8-3
 - connectivity symptoms, 5-1
 - performance symptoms, 9-6, 9-7
 - poor performance between rings, 8-7
 - poor performance when switch to
 - routing, 8-5
 - server connectivity scenario, 2-33

troubleshooting (continued)

- slow performance over Ethernet
 - Backbone, 8-9
- slow performance over parallel links, 8-13, 8-15
- other publications, 1-7
- overview, 1-1
- presenting data to technical support
 - representatives, B-3
- process defined, 1-3
- publication objectives, 1-1
- serial lines, 1-20
- specific symptoms, 1-6
- TCP/IP
 - access control scenario, 2-45
 - connectivity symptoms, 6-1
 - performance, 8-17
 - performance scenario, 8-17
 - performance symptoms, 9-8, 9-9
 - route redistribution scenario, 2-45
- third-party tools, 1-19
- Token Ring, 1-20
- tutorial information, 1-6
- WAN
 - connectivity symptoms, 7-1
 - new X.25 router connectivity scenario, 2-51
 - performance, 8-21
 - performance scenario, 8-21
 - performance symptoms, 9-11, 9-13
- worksheet, C-2

U

UNIX

- /etc/defaultrouter file, 6-4
- /etc/netmasks file, 6-4
- /etc/rc local file, 6-4
- netstat command, 6-4
- route add command, 6-4

UNIX syslog server

- logging messages to, 10-6

W

WAN

- key X.25 virtual circuit parameters, 2-59

problems

- bad access list, 7-37
- bad cables, 2-55
- bad CSU/DSU, 7-26
- buffer misses, 7-29
- carrier automatically rerouting trunk, 9-12
- clocking conflicts, 7-18
- congestion, 7-28, 8-23, 9-11
- discontinuous subnet addressing, 7-37
- faulty hardware, 7-26
- hardware and media, 8-22
- host cannot send ARPs, 7-37
- routing table errors, 7-29
- serial line congestion, 7-26, 7-27, 7-28
- specifying wrong gateway of last resort, 7-37
- timing conflicts, 7-26
- virtual circuit sequencing, 2-59
- using DCE or DTE appliques, 2-53
- See also diagnosing, symptoms, troubleshooting
- write core command, 1-9, D-1

X

X.25

- cause and diagnostics codes, A-1
- problems
 - bad cable, 7-31
 - bad router hardware, 7-31
 - dead link, 7-31
 - invalid PRs, 7-26
 - misconfigured router, 7-31
 - misconfigured switch, 7-31
- x25 encapsulation command, 5-21
- x25 map command
 - broadcast option, 2-59, 7-32
- x25 map novell command, 5-21

Z

ZIP

- diagnosing ZIP storms, 3-19
- finding ZIP storms, 2-7, 3-6
- Zone Information Protocol
 - See ZIP
- zone names, 3-8

Corporate Headquarters

Cisco Systems, Inc.
P.O. Box 3075
1525 O'Brien Drive
Menlo Park, CA 94026
USA
Tel: 415 326-1941
800 553-NETS (6387)
Fax: 415 326-1989

European Headquarters

Cisco Systems Europe, s.a.r.l.
BP 706 Evolic
16 avenue du Quebec
91961 Les Ulis Cedex
France
Tel: 33 1 6092 2000
Fax: 33 1 6928 8326
European Offices

Belgium

Tel: 32 2 643 2626
Fax: 32 2 643 2627

Germany

Tel: 49 89 3215 070
Fax: 49 89 3215 0710

Italy

Tel: 39 2 62 726 43
Fax: 39 2 62 729 13

Spain

Tel: 34 1 57 203 60
Fax: 34 1 57 071 99

Sweden

Tel: 46 8 19 62 05
Fax: 46 8 19 04 24

Switzerland

Tel: 41 55 95 60 44
Fax: 41 55 95 64 14

United Kingdom

Tel: 44 494 464944
Fax: 44 494 465300

Intercontinental Headquarters

(Latin America and Asia-Pacific)

Cisco Systems, Inc.
1525 O'Brien Drive
P.O. Box 3075
Menlo Park, CA 94026
USA

Tel: 415 326-1941
Fax: 415 688-4646

Regional Offices

Cisco Systems Australia Pty., Ltd.

Tel: 61 2 957 4944
Fax: 61 2 957 4077

Cisco Systems Canada Limited

Tel: 416 506-1500
Fax: 416 506-1506

Cisco Systems Hong Kong, Ltd.

Tel: 852 529 3534
Fax: 852 520 2676

Cisco Systems de México, S.A. de C.V.

Tel: 525 254 0880
Fax: 525 531 9659

Cisco Systems New Zealand

Tel: 9 649 358 3776
Fax: 9 649 358 4442

Japanese Headquarters

Nihon Cisco Systems K.K.
Shiba Excellent Building, 5F
2-1-13 Hamamatsucho,
MinatoKuTokyo 105, Japan
Tel: 81 3 5472 3571
Fax: 81 3 5472 3577

Cisco Systems has over 50
sales offices worldwide. Call
415 326-1941 to contact your local
account representative or, in North
America, call 800 553-NETS (6387)

