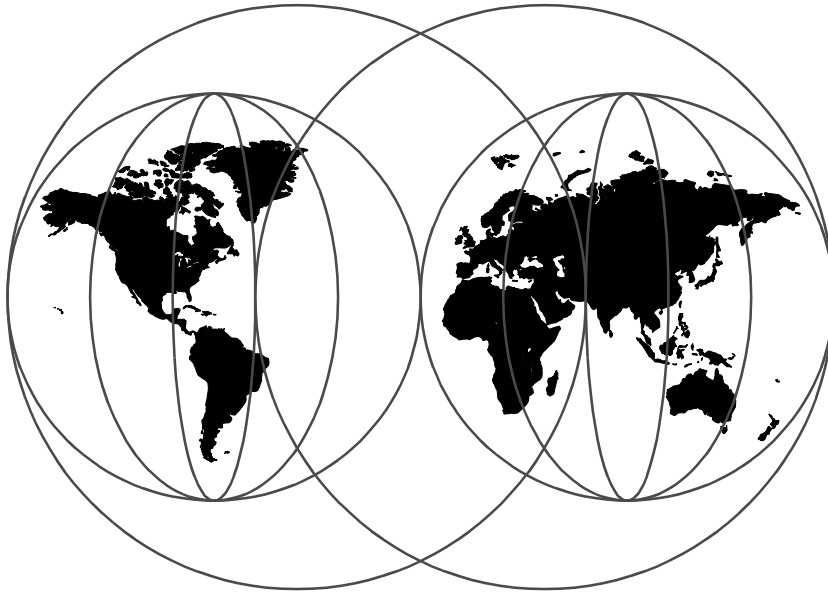


Beyond DHCP — Work Your TCP/IP Internetwork with Dynamic IP

*Peter Degotardi, Michael McDaniel, Toshimasa Shimizu
Timothy Sipples, Akiko Ueno, Uwe Zimmermann*



International Technical Support Organization

<http://www.redbooks.ibm.com>

SG24-5280-00



International Technical Support Organization

**Beyond DHCP —
Work Your TCP/IP Internetwork
with Dynamic IP**

August 1998

Take Note!

Before using the information contained in this book please be sure to read Appendix F, "Special Notices" on page 311.

First Edition (August 1998)

This edition applies to OS/2 Warp Server, OS/2 Warp 4, WorkSpace On-Demand, and TCP/IP Version 4.1 for OS/2 Warp. TCP/IP Version 4.1 for OS/2 is available from IBM Software Choice on the Internet at <http://www.software.ibm.com/swchoice>. This edition also applies to Lotus Notes, Lotus Domino, Domino Go Webserver, and Windows 3.1, Windows 95 and Windows NT client enhancements from IBM. Windows client enhancements are also available through IBM Software Choice.

Comments may be addressed to:

IBM Corporation, International Technical Support Organization
Dept. DHHB, Building 045 Internal Zip 2834
11400 Burnet Road
Austin, Texas 78758-3493

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

© Copyright International Business Machines Corporation 1998. All rights reserved

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

Contents

Figuresxi
Tablesxix
Prefacexxi
The Vision of a Perfect TCP/IP Networkxxi
How This Book Is Organizedxxii
About the Authorsxxiv
Acknowledgmentsxxvi
Comments Welcomexxviii
Redbooks Onlinexxviii
 Chapter 1. TCP/IP Basics	1
1.1 Network Protocols	1
1.2 IP Addresses	2
1.3 IP Subnets	6
1.4 IP Routing	9
1.5 Assigning IP Addresses	10
1.6 Name Servers	11
1.7 Applications That Use TCP/IP	12
1.8 Other TCP/IP Terms	12
1.9 Related Publications	14
 Chapter 2. Up and Running with DHCP	17
2.1 Why DHCP?	18
2.2 OS/2 Warp Server as a DHCP Server	18
2.2.1 Installation	18
2.2.2 DHCP Server Preparation	19
2.2.3 Basic Configuration	22
2.2.4 Testing and Verification	28
2.3 Windows NT as a DHCP Server	30
2.3.1 Installation	30
2.3.2 DHCP Server Preparation	31
2.3.3 Basic Configuration	32
2.3.4 Testing and Verification	36
2.4 DHCP Clients	37
2.4.1 OS/2 Warp 4	37
2.4.2 Windows 95	44
2.4.3 Windows NT Workstation 4.0	45
2.4.4 Apple Macintosh	47
2.4.5 IBM Network Station	49

2.4.6 Linux	55
2.4.7 DOS, Windows 3.1, and Windows for Workgroups	57
2.4.8 Sun Microsystems Solaris	64
2.4.9 IBM WorkSpace On-Demand	66
2.4.10 Hewlett-Packard LaserJet 4000 Printer (JetDirect)	69
Chapter 3. DHCP Server and Client Interaction	75
3.1 DHCP Initialization and Acquisition Process	75
3.1.1 DHCPDISCOVER	76
3.1.2 DHCPOFFER	78
3.1.3 DHCPREQUEST	80
3.1.4 DHCPACK	81
3.1.5 Verification Process	82
3.1.6 DHCPRELEASE	83
3.2 DHCP Renewing, Rebinding and Rebooting Processes	84
Chapter 4. Serving Names	87
4.1 Why Names?	87
4.2 What is DNS?	88
4.3 Domain versus Zone of Authority	91
4.4 Differentiating Name Servers	94
4.4.1 Static Name Servers	94
4.4.2 Dynamic Name Servers	94
4.4.3 Primary Name Servers	95
4.4.4 Secondary Name Servers	95
4.4.5 Master Name Servers	96
4.4.6 Caching-Only Name Servers	96
4.4.7 Authoritative Name Servers	96
4.4.8 Parent and Child Name Servers	97
4.4.9 Root Name Servers	97
4.4.10 Forwarders	98
4.4.11 Firewall Name Servers	98
4.4.12 Record Types	99
4.5 Windows NT as a Static DNS Server	100
4.5.1 Scenario	100
4.5.2 Tasks	101
4.5.3 Planning	101
4.5.4 Setting Up	102
4.6 DNS Client Support in Windows NT	111
4.6.1 Windows 95	111
4.6.2 Windows NT Workstation 4.0	112
4.7 OS/2 Warp Server as Dynamic DNS Server	112
4.7.1 The Dynamic Domain Name System (DDNS)	112

4.7.2	IBM's Dynamic DNS Relative to Other Implementations	118
4.7.3	Scenario	121
4.7.4	Tasks	122
4.7.5	Planning	122
4.7.6	Setting Up your DDNS Server	123
4.7.7	ProxyArec	137
4.7.8	Verification	138
4.7.9	DDNS Files	138
4.7.10	Providing Mail Services	146
4.8	Dynamic DNS Client Support in OS/2 Warp Server	158
4.8.1	OS/2 Warp 4 with TCP/IP Version 4.1 for OS/2	158
4.8.2	OS/2 Warp 4 with TCP/IP 4.0	161
4.8.3	Windows 95/NT with IBM Dynamic IP Client	163
4.8.4	OS/2 Warp 4 Using Presecured Mode	168
4.8.5	Windows 95 Using Presecured Mode	171
4.8.6	Windows 95 Using Proxy	174
4.8.7	WorkSpace On-Demand Using Proxy	174
4.9	Related Publications	176
Chapter 5. Integrating File and Print Services		177
5.1	NetBIOS, NetBEUI and TCPBEUI: What Are They?	177
5.2	Resolving NetBIOS Names to IP Addresses	181
5.2.1	Enhancing B-Node Clients	182
5.3	New Way of Using DNS with Non-RFC-Encoded Name	195
5.4	NetBIOS Name Servers (NBNS)	201
5.4.1	Microsoft WINS	201
5.4.2	Network TeleSystems Shadow IPserver	203
5.5	P-Node, H-Node and M-Node Clients	209
5.5.1	Manually Configuring Clients to Use a NBNS	209
5.5.2	Dynamically Configuring Clients to Use a NBNS	214
5.6	IBMs Neighborhood Browser Enabler for Warp Server	222
5.6.1	Overview	222
5.6.2	Installing Neighborhood Browser Enabler	223
5.6.3	Starting and Stopping the Neighborhood Browser Enabler	224
5.7	Related Publications	224
Chapter 6. Growing Your Network		227
6.1	Multiple Subnets	227
6.1.1	Static and Dynamic Routing Between Subnets	227
6.1.2	OS/2 Warp as a Router	229
6.1.3	Windows NT as a Router	232
6.2	Crossing Routers	235
6.2.1	Windows NT as a DHCP Relay Agent	236

6.2.2 OS/2 Warp as a DHCP Relay Agent	239
6.2.3 DHCP Relay Considerations	240
6.3 IP Masquerading / Network Address Translation	241
6.3.1 Translation Mechanism	243
6.4 DHCP and DDNS with UNIX	243
6.4.1 AIX as a DHCP Server	244
6.4.2 AIX as a DDNS Server	258
6.5 DHCP and DDNS with Shadow IPserver	258
6.5.1 Configuration Files and Save Files	259
6.5.2 Using Configuration Files	260
6.5.3 Configuring through IPmanager	269
Chapter 7. Mobile Users	279
7.1 PPP Dial-Up	279
7.1.1 Prerequisites	280
7.1.2 Step-by-Step Procedure	282
7.2 Roaming Users	284
7.2.1 Prerequisites	285
7.2.2 Automatic Connection to File/Print Domain	285
7.2.3 Netscape "Message of the Day" Service	295
7.2.4 Custom Netscape INI Files	302
7.2.5 Summary	307
Chapter 8. Security of DHCP and Dynamic DNS	309
8.1 Security Trade-off	309
8.2 RSA Public Key Authentication System	310
8.3 Getting More Information from the Client to DNS	313
8.4 Presecured Domain	315
8.5 ProxyArec Consideration	318
8.5.1 ProxyArec and Option 81	320
8.6 Securing Lease Allocations	322
8.6.1 Preventing Access to Unauthorized Devices	322
8.6.2 'Rogue' DHCP Servers	322
8.6.3 Connecting to Untrusted Networks - Firewalls	323
8.6.4 Connecting Through Untrusted Networks - VPN	325
8.6.5 TFTP Security	326
Chapter 9. Reliability	327
9.1 Battlefield Questions	327
9.2 Failure Events	328
9.2.1 Severed Connections	328
9.2.2 Facility Loss	328
9.2.3 Router Outages	329
9.2.4 DHCP Server Problems	329

9.2.5	Name Server Difficulties	329
9.2.6	Other Server Vulnerabilities	330
9.2.7	Client Failures	331
9.3	A "Fault Tolerant" DHCP Server	331
9.3.1	Prerequisites	332
9.3.2	Step-by-Step Procedure	332
9.3.3	Technical Caveats	338
9.4	AIX Features	340
9.5	Shadow IPserver Features	340
Chapter 10. Performance		341
10.1	Leases	341
10.1.1	What is a Lease?	341
10.1.2	How Leases Work.	341
10.1.3	Choosing a Lease Time	342
10.1.4	Multiple Leases	343
10.2	Monitoring and Troubleshooting	344
10.2.1	The PING Command	344
10.2.2	The TRACERTE Command	345
10.2.3	The IPTRACE Command	346
10.2.4	The ARP Command	346
10.2.5	The NETSTAT Command	347
10.2.6	The HOST Command	348
10.2.7	The NSLOOKUP Command	348
10.2.8	Other Utilities	349
10.3	Troubleshooting TCP/IP Networks	350
10.3.1	Prerequisites for Troubleshooting	350
10.3.2	A Bottom-Up Approach	350
10.4	Tuning TCP/IP Networks	361
10.4.1	An Approach to Tuning Your Network	362
10.4.2	TCP/IP Tuning Parameters	363
10.5	Bandwidth Efficiency	366
10.5.1	Broadcast Traffic	366
10.5.2	RSVP	367
10.5.3	Communications Server	367
10.6	Related Publications	368
Chapter 11. Managing Your Network		369
11.1	Remote Administration of DHCP and DDNS	369
11.1.1	Look! No Extra Software	369
11.1.2	Using a Web Server	375
Appendix A. Creating DHCP Boot Diskettes		385
A.1	OS/2 Warp Boot Diskettes	385

A.1.1 Prerequisites	385
A.1.2 Step-by-Step Procedure	386
A.1.3 Notes on the Step-by-Step Procedure	394
A.2 DOS Boot Diskettes	395
A.2.1 Prerequisites	395
A.2.2 Step-by-Step Procedure	395
A.2.3 Notes on the Step-by-Step Procedure	400
A.3 Related Publications	400
Appendix B. Where Is It? Internet and IBM Intranet Web Sites	401
B.1 Internet Web and FTP Sites	401
B.2 IBM Intranet Web Sites	402
Appendix C. Application Issues	405
C.1 DB2 Universal Database	405
C.2 TME 10 Framework	406
C.3 NetFinity	406
C.3.1 Manager	407
C.3.2 Client	408
C.4 Netscape Navigator	410
Appendix D. CD-ROM Contents	411
Appendix E. DHCP Options (RFC 2132)	413
E.1 Introduction	413
E.2 DHCP and BOOTP Options	414
E.2.1 Options 0 and 255: Pad and End	414
E.2.2 Option 1: Subnet Mask	414
E.2.3 Option 2: Time Offset	414
E.2.4 Option 3: Router	415
E.2.5 Option 4: Time Server	415
E.2.6 Option 5: IEN 116 (Old) Name Server	415
E.2.7 Option 6: Domain Name Server	416
E.2.8 Option 7: Log Server	416
E.2.9 Option 8: Cookie Server	416
E.2.10 Option 9: LPR Server	416
E.2.11 Option 10: Impress Server	417
E.2.12 Option 11: Resource Location Server	417
E.2.13 Option 12: Host Name	417
E.2.14 Option 13: Boot File Size	418
E.2.15 Option 14: Merit Dump File Name	418
E.2.16 Option 15: Domain Name	418
E.2.17 Option 16: Swap Server	418
E.2.18 Option 17: Root Path	419

E.2.19	Option 18: Extensions Path	419
E.2.20	Option 19: IP Forwarding Enable/Disable	419
E.2.21	Option 20: Non-Local Source Routing Enable/Disable	419
E.2.22	Option 21: Policy Filter	420
E.2.23	Option 22: Maximum Datagram Reassembly Size	420
E.2.24	Option 23: Default IP Time-to-Live	421
E.2.25	Option 24: Path MTU Aging Timeout	421
E.2.26	Option 25: Path MTU Plateau Table	421
E.2.27	Option 26: Interface MTU	421
E.2.28	Option 27: All Subnets Are Local	422
E.2.29	Option 28: Broadcast Address	422
E.2.30	Option 29: Perform Mask Discovery	422
E.2.31	Option 30: Mask Supplier	423
E.2.32	Option 31: Perform Router Discovery	423
E.2.33	Option 32: Router Solicitation Address	423
E.2.34	Option 33: Static Route	423
E.2.35	Option 34: Trailer Encapsulation	424
E.2.36	Option 35: ARP Cache Timeout	424
E.2.37	Option 36: Ethernet Encapsulation	424
E.2.38	Option 37: TCP Default Time-to-Live	425
E.2.39	Option 38: TCP Keepalive Interval	425
E.2.40	Option 39: TCP Keepalive Garbage	425
E.2.41	Option 40: Network Information Service Domain	426
E.2.42	Option 41: Network Information Server	426
E.2.43	Option 42: Network Time Protocol Server	426
E.2.44	Option 43: Vendor-Specific Information	426
E.2.45	Option 44: NetBIOS over TCP/IP Name Server Option	427
E.2.46	Option 45: NetBIOS over TCP/IP Datagram Distribution Server	428
E.2.47	Option 46: NetBIOS over TCP/IP Node Type	428
E.2.48	Option 47: NetBIOS over TCP/IP Scope	428
E.2.49	Option 48: X Window System Font Server Option	429
E.2.50	Option 49: X Window System Display Manager	429
E.2.51	Option 64: Network Information Service (Plus) Domain	429
E.2.52	Option 65: Network Information Service (Plus) Server	430
E.2.53	Option 68: Mobile IP Home Agent	430
E.2.54	Option 69: Simple Mail Transport Protocol (SMTP) Server	430
E.2.55	Option 70: Post Office Protocol (POP3) Server	430
E.2.56	Option 71: Network News Transport Protocol (NNTP) Server	431
E.2.57	Option 72: Default World Wide Web (WWW) Server	431
E.2.58	Option 73: Default Finger Server	431
E.2.59	Option 74: Default Internet Relay Chat (IRC) Server	432
E.2.60	Option 75: StreetTalk Server	432
E.2.61	Option 76: StreetTalk Directory Assistance (STDA) Server	432

E.3 DHCP (Only) Options	432
E.3.1 Option 50: Requested IP Address	433
E.3.2 Option 51: IP Address Lease Time	433
E.3.3 Option 52: Option Overload	433
E.3.4 Option 53: DHCP Message Type	434
E.3.5 Option 54: Server Identifier	434
E.3.6 Option 55: Parameter Request List	434
E.3.7 Option 56: Message	435
E.3.8 Option 57: Maximum DHCP Message Size	435
E.3.9 Option 58: Renewal (T1) Time Value	435
E.3.10 Option 59: Rebinding (T2) Time Value	436
E.3.11 Option 60: Vendor Class Identifier	436
E.3.12 Option 61: Client Identifier	436
E.3.13 Option 66: TFTP Server Name	437
E.3.14 Option 67: Boot File Name	437
E.4 Unofficial DHCP Options	437
E.5 Options Supported by Popular Operating Systems	439
E.5.1 Servers	439
E.5.2 Clients	439
Appendix F. Special Notices	441
Appendix G. Related Publications	445
G.1 International Technical Support Organization Publications	445
G.2 Redbooks on CD-ROMs	445
G.3 Other Publications	446
G.4 Request for Comments (RFC)	446
How To Get ITSO Redbooks	449
How IBM Employees Can Get ITSO Redbooks	449
How Customers Can Get ITSO Redbooks	450
IBM Publications Order Form	451
Glossary and Abbreviations	453
Index	463
ITSO Redbook Evaluation	471

Figures

1. IP Classes	3
2. DHCP Cycle	17
3. [Warp Server] TCP/IP Version 4.1 Configuration Folders	20
4. [Warp Server] TCP/IP Administrator Password Window	20
5. [Warp Server] TCP/IP Configuration Notebook	21
6. [Warp Server] DHCP Server Services	22
7. [Warp Server] Password for DHCP Server Configuration	23
8. [Warp Server] DHCP Server Sample Configuration	24
9. [Warp Server] DHCP Server New Configuration	25
10. [Warp Server] DHCP Subnet Notebook	26
11. [Warp Server] DHCP Server Initialized	27
12. [Warp Server] DHCP Log File Notebook Settings Notebook	28
13. [Windows NT] Creating a Scope	34
14. [Windows NT] Adding DHCP Options	34
15. [Windows NT] Entering Value for Option 15	35
16. [Windows NT] Creating Address List for Option 3	35
17. [Windows NT] DHCP Active Leases	36
18. [Windows NT] DHCP Client Properties	37
19. [OS/2 Warp] Enabling a DHCP Client with TCP/IP 4.1	40
20. [OS/2 Warp] DHCP Client Monitor Using TCP/IP 4.1	41
21. [OS/2 Warp] Enabling a DHCP Client Using TCP/IP 4.0	42
22. [OS/2 Warp] DHCP Client Monitor Window	43
23. [OS/2 Warp] Current Configuration Window	43
24. [Windows 95] Enabling DHCP Client	44
25. [Windows 95] IP Configuration Window	45
26. [Windows NT] Enabling a DHCP Client	46
27. [Windows NT] Displaying Current Configuration	47
28. [Macintosh] Selecting the TCP/IP Control Panel	48
29. [Macintosh] Choosing the DHCP Server Option	48
30. [Macintosh] Saving TCP/IP Configuration Changes	49
31. [Warp Server] TCP/IP Configuration Notebook: Autostart Service	50
32. [Warp Server] TCP/IP Configuration Notebook: Syslog Daemon	51
33. [Warp Server] Client Definition Settings Notebook	52
34. [Warp Server] DHCP Options Settings Notebook	53
35. [Warp Server] Support BOOTP Settings Notebook	54
36. [RedHat Linux] Control Panel	56
37. [RedHat Linux] Interfaces Section of Network Configuration	56
38. [RedHat Linux] Editing the tr0 (Token-Ring) Interface	57
39. [DOS LAN Services] Installation Options	58
40. [DOS LAN Services] Selecting a Protocol	59

41. [DOS LAN Services] Error When Required Value Not Specified	60
42. [DOS LAN Services] Options for the Protocol Driver	61
43. [DOS LAN Services] Network Adapter Types	62
44. [DOS LAN Services] Extract from NETWORK.INI	63
45. [DOS LAN Services] Messages Displayed as a DOS Workstation Starts .	63
46. [DOS LAN Services] Testing IP Configuration Using PING	64
47. Integrating the Sun Solaris Workstation into Our Network	65
48. DHCP Results on Sun UltraSparc	66
49. [WorkSpace On-Demand] Contents of Remote IPL Requesters Folder . .	67
50. [WorkSpace On-Demand] IP Address Setup for Client	68
51. [Warp Server] DHCP Printer Configuration	70
52. Executable Commands at the HP Printer Through a Telnet Session . . .	71
53. HP Printer Configuration after Boot Up	72
54. [OS/2 Warp] Create LPD Printer	73
55. [OS/2 Warp] LPD Settings Notebook	74
56. DHCP Client and DHCP Server Interaction	75
57. [{NetXRay] DHCPDISCOVER Packet	76
58. [NetXRay] Packets from DHCP Relay Agent to DHCP Server	77
59. [NetXRay] Broadcast Packets	78
60. [NetXRay] ARP Packets	79
61. [NetXRay] DHCP OFFER Packet	80
62. [NetXRay] DHCP REQUEST Packet	81
63. [NetXRay] DHCP ACK Packet	82
64. [NetXRay] Verify State (ARP)	83
65. [NetXRay] DHCP RELEASE Packet	84
66. DNS Name Space	89
67. Hosts with the Same Names in Different Domains	91
68. Domain, Subdomain, Delegation, and Zone of Authority	94
69. [Windows NT] Adding DNS Server	103
70. [Windows NT] Creating a New Zone for Forward Mappings	104
71. [Windows NT] Creating a New Zone for Reverse Mappings	104
72. [Windows NT] Adding a New Host	105
73. [Windows NT] DHCP – WINS – DNS Interaction	106
74. [Windows NT] Enabling WINS Lookup	108
75. [Windows NT] Enabling WINS Reverse Lookup	108
76. [Windows NT] BELLEVUE.COOKING.NET.DNS File	110
77. [Windows NT] 8.168.192.IN.ADDR.ARPA.DNS File	111
78. [Warp Server] DHCP – DDNS – Client Interaction	114
79. [Warp Server] Dynamic Presecured Domain	117
80. [Warp Server] Using ProxyArec	118
81. [Warp Server] HOSTS File	124
82. [Warp Server] Domain Name Server Notebook	125
83. [Warp Server] Primary Domain Configuration	126

84. [Warp Server] Setting Secured Mode	126
85. [Warp Server] DDNS Server Administrator Notebook	127
86. [Warp Server] Adding Hosts	128
87. [Warp Server] Specifying DDNS Server for PTR Record Update	129
88. [Warp Server] Enabling DHCP Server to Update PTR Records	130
89. [Warp Server] Domain Name Server Notebook	131
90. [Warp Server] Primary Domain Configuration Notebook	132
91. [Warp Server] Setting Dynamic Presecured Mode	132
92. [Warp Server] DDNS Server Administrator Notebook	133
93. [Warp Server] Adding Hosts	134
94. [Warp Server] Generate Keys for DDNS Clients	135
95. [Warp Server] Specifying DDNS Server for PTR Record Update	136
96. [Warp Server] Enabling DHCP Server to Update PTR Records	137
97. [Warp Server] DNSEXT.CFG	140
98. [Warp Server] NAMED.BT File	145
99. [Warp Server] DNSF0000.DOM File	146
100.[Warp Server] DNSF0000.STA File	146
101.[OS2POPS] Installation Panel	148
102.[OS2POPS] Client Maintenance	148
103.[OS2POPS] Manually Modifying SENDMAIL.CF	149
104.[Warp Server] Setting Sendmail to Autostart.	150
105.[Warp Server] Modifying Global DHCP Parameters	151
106.[Warp Server] Defining an SMTP Server	152
107.[Warp Server] Defining a POP3 Server	153
108.[Warp Server] Adding a Mail Exchange to the Primary Domain	154
109.[Warp Server] Modification to Domain File Showing New MX Record . .	155
110.[Warp Server] NETSTAT -s Showing SMPT and POP3 Sockets	156
111.[Netscape] Mail Server Configuration	157
112.[Netscape] Reading Email	157
113.[OS/2 Warp] Enabling the DDNS Client	159
114.[OS/2 Warp] DDNS Client Configuration	160
115.[OS/2 Warp] Selecting a Previous Configuration	161
116.[OS/2 Warp] Enabling the DDNS Client	162
117.[OS/2 Warp] DDNS Client Configuration	163
118.[OS/2 Warp] DHCP Monitor	163
119.[Windows NT] Dynamic IP Client Configuration	165
120.[Windows NT] Dynamic IP Client Diagnostics Tab	165
121.[Windows 95] Dynamic IP Client Configuration	167
122.[Windows 95] Dynamic IP Client Diagnostics Tab	167
123.[OS/2 Warp] Enabling the DDNS Client	169
124.[OS/2 Warp] Importing Key File	170
125.[OS/2 Warp] DDNS Configuration Message	170
126.[OS/2 Warp] DDNS Client Configuration	171

127.[Windows 95] Registry Editor Message	173
128.[WSOD] TCP/IP Setting Notebook	175
129.[WSOD] DHCP Client Monitor	176
130.Workstation Communicating Using Multiple Protocols	178
131.[OS/2 Warp] NetBIOS, NetBEUI and TCPBEUI Structure	179
132.[DOS LAN Services] Example Input File for NBUTIL	184
133.[DOS LAN Services] Example AUTOEXEC.BAT Using NBUTIL	184
134.[DOS LAN Services] Sample Output from NBUTIL	185
135.[Windows 95/NT] Sample LMHOSTS File.	187
136.[Windows NT] Enabling Use of LMHOSTS	187
137.[OS/2 Warp] NetBIOS Over TCP/IP Configuration Options	189
138.[OS/2 Warp] Modifying RFCNAMES.LST through MPTS	189
139.[OS/2 Warp] Modifying RFCBCST.LST through MPTS	191
140.[Warp Server] Sample DNS Database	193
141.[Warp Server] DNS Database with Encoded NetBIOS Names.	194
142.Using DNS Server with Node Status Request	195
143.[Windows NT] WINS Address No Use of WINS but Use DNS	197
144.[Windows NT] TCP/IP Property - DNS Settings	198
145.[Windows NT] Packet Trace - Node Status Request	199
146.[Windows NT] NBTSTAT Output.	200
147.[Shadow IPServer] NTS-SRVR.CFG Configuration File	207
148.[Shadow IPserver] Sample NTS-SRVR.NBN file	209
149.[DOS LAN Services] PROTOCOL.INI Extract Showing NBNS Setup. . .	210
150.[Windows 95] Manually Configuring a NBNS	211
151.[Windows NT] Manually Configuring a NBNS	212
152.[OS/2 Warp] Manually Configuring a NBNS	213
153.SETNBNS.CMD (Part 1 of 3)	215
154.SETNBNS.CMD (Part 2 of 3)	216
155.SETNBNS.CMD (Part 3 of 3)	217
156.GETNBNS.CMD (Part 1 of 4)	218
157.GETNBNS.CMD (Part 2 of 4)	219
158.GETNBNS.CMD (Part 3 of 4)	220
159.GETNBNS.CMD (Part 4 of 4)	221
160.Sample Network for Routing	228
161.[OS/2 Warp] Enabling IP Forwarding	229
162.[OS/2 Warp] Adding a Route	230
163.[OS/2 Warp] Enabling ROUTED	232
164.[Windows NT] Enabling IP Forwarding	233
165.[Windows NT] Installing the RIP for Internet Protocol Service	235
166.DHCP Initialization through a DHCP Relay Agent	236
167.[Windows NT] List of Available Network Services.	237
168.[Windows NT] Error Panel Shown after Relay Agent Install	237
169.[Windows NT] DHCP Relay Tab	238

170.[Windows NT] Adding a DHCP Server	238
171.[Windows NT] Reboot Prompt	238
172.[OS/2 Warp] Sample DHCP.D.CFG File	239
173.Sample Environment	245
174.[AIX] DHCP Server Configuration	246
175.[AIX] Adding a DHCP Server	247
176.[AIX] Choosing a DHCP Server	247
177.[AIX] Adding a Domain Name	248
178.[AIX] Adding a DNS	248
179.[AIX] Adding a Network	249
180.[AIX] Adding a Subnet	250
181.[AIX] Adding a Default Router	250
182.[AIX] Completed Server Configuration	251
183.[AIX] Final Configuration	252
184.[AIX] Saving the Master File with [File – Save As]	252
185.[AIX] Saving an Individual Server file with [Server – Save File for]	253
186.[AIX] The masterfile.cnf File (Part 1 of 2)	254
187.[AIX] The masterfile.cnf File (Part 2 of 2)	255
188.[AIX] The dhcpsd.cnf File	256
189.[Shadow IPserver] NTS-SRVR.DHC File	267
190.[Shadow IPserver] NTS-SRVR.DNS File	269
191.[Shadow IPmanager] Logon Panel	270
192.[Shadow IPmanager] Startup Screen	270
193.[Shadow IPmanager] DHCP Option Sets	271
194.[Shadow IPmanager] Editing a DHCP Option Set	271
195.[Shadow IPmanager] Adding DHCP Options to a Pool	272
196.[Shadow IPmanager] DHCP Pools	273
197.[Shadow IPmanager] Editing a DHCP Pool	273
198.[Shadow IPmanager] DHCP Configurations	274
199.[Shadow IPmanager] Editing a DHCP Station	275
200.[Shadow IPmanager] DNS Entries	276
201.[Shadow IPmanager] Adding a DNS Entry	276
202.[Shadow IPmanager] The Options Menu	277
203.[Shadow IPmanager] Importing Server Databases	277
204.[Shadow IPmanager] Converting Save Files to Text	278
205.[Warp Server] PPP Server and DHCP	280
206.[Warp Server] PPP Server and DHCP: Example	281
207.[Warp Server] TCP/IP Added to PPP Logical Adapter	282
208.[Warp Server] IP Forwarding Enabled	283
209.[Warp Server] Adding Class Definition to DHCP Server	286
210.[Warp Server] Adding Option 150 to Class IBMWARP_V3.1	287
211.[Warp Server] Setting ARMONK as DHCP Option 150 Value	288
212.[Warp Server] Option Values for Class IBMWARP_V.3.1	289

213.[OS/2 Warp] MYOFFICE.CMD (Part 1 of 3)	290
214.[OS/2 Warp] MYOFFICE.CMD (Part 2 of 3)	291
215.[OS/2 Warp] MYOFFICE.CMD (Part 3 of 3)	292
216.[OS/2 Warp] STARTUP.CMD	293
217.[OS/2 Warp] MYWEBMSG.CMD	298
218.[OS/2 Warp] Program Object in Startup Folder (Page 1)	299
219.[OS/2 Warp] Program Object in Startup Folder (Page 2)	299
220.[OS/2 Warp] SEEMYMSG.CMD	300
221.[OS/2 Warp] Netscape in Kiosk Mode	301
222.[OS/2 Warp] NSSOCKS.CMD (Part 1 of 2)	304
223.[OS/2 Warp] NSSOCKS.CMD (Part 2 of 2)	305
224.RSA Public Key Authentication System	310
225.[Warp Server] Example of KEY, SIG and A Resource Record	311
226.[Warp Server] NSLOOKUP Output Returning Public Key	313
227.[Warp Server] DHCP Server, Configuring Option 192 TXT Record	314
228.[Warp Server] DDNS Client Prompts for the Input	315
229.Presecured TCP/IP Domain Concept	316
230.Combination of Static DNS and Dynamic Presecured DNS	317
231.[Warp Server] ProxyArec - Enable DHCP Server to Update A RR	319
232.[Warp Server] Enable ProxyArec and Option for Protection	319
233.[OS/2 Warp] DHCPD.CFG Example with Option 81	321
234.IETF Definition of Option 81	321
235.[Warp Server] Choosing Autostart Service for REXECD	333
236.[Warp Server] Security Section in TCP/IP Configuration	334
237.[Warp Server] Configuring REXECD Options	335
238.[OS/2 Warp] FAILSAFE.CMD (Part 1 of 2)	336
239.[OS/2 Warp] FAILSAFE.CMD (Part 2 of 2)	337
240.Output from NETSTAT -i	354
241.Output from NETSTAT -t	355
242.Name Resolution Example	357
243.RDDNSAPC.CMD (Part 1 of 2)	371
244.RDDNSAPC.CMD (Part 2 of 2)	372
245.RDADMGUI.CMD (Part 1 of 2)	372
246.RDADMGUI.CMD (Part 2 of 2)	373
247.RDHCPSCP.CMD (Part 1 of 2)	374
248.RDHCPSCF.CMD (Part 2 of 2)	375
249.[Go Webserver] Configuration and Administration Forms	378
250.[Go Webserver] Request Routing Page	379
251.[Go Webserver] Insert Routing Request	380
252.[OS/2 Warp] Remote Configuration of TCP/IP Version 4.1 for OS/2	381
253.[NetFinity] NetFinity Service Manager Folder	407
254.[NetFinity] System Group Management Options	407
255.[NetFinity] Dynamic Address Options	408

256.[NetFinity] Network Driver Configuration for a Client.	408
257.[NetFinity] Network Driver Configuration Updated	409
258.[OS/2 Warp] DDNS Client Configuration.....	410
259.CD-ROM Contents: INDEX.HTM	411

Tables

1. Class Determination Table.	5
2. Class Properties.	6
3. Number of Subnets and Hosts per Subnet (Partial).	8
4. Windows NT Server Configuration Information	101
5. DNS Configuration Files	109
6. OS/2 Warp Server Configuration Information	121
7. DNSEXT.CFG Configuration Parameters	141
8. NetBIOS Names and Suffixes	180
9. WINS Hardware and Software Requirements	202
10. Shadow Hardware and Software Requirements	205
11. Routing Tables Required for Sample Network.	228
12. NTS-SRVR.CFG Global System Keywords.	261
13. NTS-SRVR.CFG Global NIC Keywords.	261
14. NTS-SRVR.CFG Global IP Keywords	262
15. NTS-SRVR.CFG Global DHCP Parameters	262
16. NTS-SRVR.CFG Global DNS Keywords	263
17. NTS-SRVR.CFG Global NBNS Parameters	264
18. NTS-SRVR.DHC General Keywords	265
19. NTS-SRVR.DHC DHCP Option Keywords	266
20. Routing Request Table Entry	380

XX Beyond DHCP — Work Your TCP/IP Internetwork with Dynamic IP

Preface

Beyond DHCP — Work Your TCP/IP Internetwork with Dynamic IP explores important network design issues for today's modern mixed intranets. The authors discuss many different platforms, including Windows 95, Windows NT, OS/2 Warp, OS/2 Warp Server, AIX, Macintosh, WorkSpace On-Demand, network computers, Linux, Solaris, and others. The book examines how to connect these systems in a reliable, flexible, high performance TCP/IP network.

When confronted with thorny TCP/IP problems, network specialists, support staff, network managers, and other technicians can refer to this book to troubleshoot network outages and to prevent them from recurring. The authors emphasize cost effective remote management and dynamic setup, providing real world examples of DHCP and DDNS technologies applied to roaming users, intelligent Web browsing, software distribution, printing, PPP dial-up, and other network needs.

This publication helps small TCP/IP network operators establish a solid foundation for future growth without undue expense. Enterprise network staff will appreciate the sections exploring high end technologies, including Network TeleSystems' Shadow IPserver, IBM Communications Server, and UNIX platforms. All network managers should find the authors' advice on security well worth reading.

The Vision of a Perfect TCP/IP Network

Why are computer networks, especially TCP/IP networks, so much harder to plug into than, say, the telephone network?

We have a vision of a more perfect TCP/IP network. The network should:

1. Allow people to connect any kind of computing device to the network and be instantly and effortlessly connected to the information they need.
2. Assure that a connection will always be available.
3. Provide top level performance even on connections with limited speed.
4. Protect private information from unwanted access.
5. Grow and change quickly and easily, keeping up with changes in the marketplace and new business demands.
6. Assure ease of administration.

Does your network perform this way?

If you (or your boss) share some or all of this vision, and you're looking for the benefits it provides, this book is for you. We'll show you how to design a top class TCP/IP network, public or private, which works a lot more like the telephone network. You'll be able to plug printers, PCs, Macintoshes, network computers, servers, and other devices into your network with little or no configuration. People carrying notebook computers from office to office will be able to plug in, and not only get access to the network, but also get access to the right printers, LANs, and applications wherever they happen to be working. If 100 new employees arrive on Monday morning, you'll be able to give them access to the network without having to drive to the office. If 100 employees depart the company for good on Friday, you'll be able to prevent any bad apples from even connecting. You'll reduce or eliminate delays caused by clogged network connections. As a result, you'll get back a lot of lost time (and money). Finally, if there is a network problem, you'll be able to fix it now, rather than next week, from anywhere you happen to be, and you might even fix it before anyone else knows about it.

We'll go beyond simple DHCP (Dynamic Host Configuration Protocol) to enable you to deliver a responsive and reliable TCP/IP network. We'll show you how to build this network from scratch or, if you prefer, how to make simple additions to your existing network to deliver on this vision. We promise to focus on the benefits and not simply rattle through a set of product marketing specifications; so you can pick and choose the solutions that make sense for you. Then, you'll also be able to compare various products to help choose the ones that best solve your problems.

Armed with the knowledge you'll get in this book, you'll look like a hero to the people who use your TCP/IP network. We hope you enjoy your promotion!

How This Book Is Organized

Feel free to skip around in this book depending on your network needs. Chapters include:

1. **TCP/IP Basics.** Chapter 1, "TCP/IP Basics" on page 1 reviews basic TCP/IP terminology and concepts. If you're an expert in subnets, routers, and all the acronyms, you can skim through this chapter.
2. **Up and Running.** In Chapter 2, "Up and Running with DHCP" on page 17 we'll show how to set up a simple TCP/IP network with one subnet. Each system connected to this simple network will be provided address, subnet mask, router address, and name server information to provide the most

basic connection. These critical network connection values will be provided using DHCP, and we'll explain why having dynamic addressing helps your network run more smoothly. Discover how to connect just about any type of computing device to your DHCP network for true interoperability.

3. **DHCP Technical Primer.** If you want to examine network traces and get more information on DHCP conversations across the network, take a look at Chapter 3, "DHCP Server and Client Interaction" on page 75.
4. **Serving Names.** Chapter 4, "Serving Names" on page 87 reveals how to assign unique names, dynamically, to all the devices on your TCP/IP network. See how a dynamic name server works and how it integrates with existing static name servers and firewalls. We'll also discuss how to choose a naming scheme and whether your systems should have (or even need) a TCP/IP name.
5. **Integrating File and Print Services.** Chapter 5, "Integrating File and Print Services" on page 177 discusses how to integrate existing LANs into your TCP/IP network. This chapter focuses on legacy protocols (such as NetBEUI), NetBIOS name servers, and Microsoft WINS. If you are interested in integrating printing and file sharing into your TCP/IP network, you should read this chapter carefully.
6. **Growing Your Network.** We discuss issues relating to multiple subnets in Chapter 6, "Growing Your Network" on page 227. Should multiple DHCP servers be used? How should address pools be managed? We'll also show how dynamic addressing can be used even when routers do not handle such traffic. If your network is growing, you'll want to read this chapter.
7. **Mobile Users.** If you're planning on supporting remote users or offices, Chapter 7, "Mobile Users" on page 279 will help explain PPP with dial-up or ISDN connections. We'll also demonstrate how to automatically assign printers and display welcome messages, using Netscape, as users roam from office to office.
8. **Security of DHCP and DDNS.** Security is often an afterthought in building a network, but you should be very concerned that your private (or even public) TCP/IP network can prevent unwanted access. In Chapter 8, "Security of DHCP and Dynamic DNS" on page 309 we'll show how to refuse connections to the network; so no one with a notebook computer can walk in and grab your information (unless you want them to).
9. **Reliability.** Chapter 9, "Reliability" on page 327 focuses on how to make your TCP/IP network super reliable, with connections always available

and conflicts eliminated. We'll explain how to create fault tolerant TCP/IP address servers.

10. **Performance.** If you're looking to boost the performance and capacity of your TCP/IP network, Chapter 10, "Performance" on page 341 should help. This chapter examines how to set lease times and minimize broadcast traffic, among other issues.
11. **Managing Your Network.** Chapter 11, "Managing Your Network" on page 369 provides insight into how to respond to changes in your network quickly and easily. Read how to make changes to your TCP/IP network from any location, adding or deleting new users and connections as needed.

In the appendices, you will find some specialized information:

- **Creating DHCP Boot Diskettes.** Appendix A, "Creating DHCP Boot Diskettes" on page 385 examines how to create bootable DOS and OS/2 Warp diskettes, which work with your dynamic TCP/IP network; so you can install Windows or just about any other kind of software to new PCs.
- **Where Is It?** Appendix B, "Where Is It? Internet and IBM Intranet Web Sites" on page 401 provides some Internet and IBM intranet Web sites for you to use in acquiring extra software described in this book (for example, the latest Java, TCP/IP, fixes, and so on).
- **Application Issues.** Appendix C, "Application Issues" on page 405 describes application issues (DB2, NetFinity, Netscape, and so on) and other problem solutions.
- **CD-ROM Contents.** Appendix D, "CD-ROM Contents" on page 411 lists the contents of the CD-ROM accompanying this book.
- **DHCP Options.** Appendix E, "DHCP Options (RFC 2132)" on page 413 provides a full list of all DHCP options, with descriptions.

About the Authors

This book was produced by a team of specialists from around the world working at the IBM International Technical Support Organization (ITSO) Center in Austin, Texas:

- **Uwe Zimmermann** is an Advisory Systems Engineer and Project Leader at the ITSO Austin Center. He has more than ten years of experience in heterogeneous networking environments. Before joining the ITSO, Uwe worked in an IBM branch office in Stuttgart, Germany, and was in charge of large accounts as a Networking Systems Engineer. His areas of expertise include OS/2 LAN/Warp Server, Windows 95/NT, NetWare,

software distribution (CID), systems management, Dynamic IP, and network computing.

- **Peter Degotardi** is a LAN Systems Specialist within IBM Global Services Australia. He has twelve years of experience in networking with a wide variety of computing devices and operating systems. His current areas of expertise include OS/2 Warp, OS/2 LAN/Warp Server, Windows NT, TCP/IP, NetFinity, REXX, and software distribution via CID.
- **Michael McDaniel**, a member of ACM, CPSR, and IEEE, and is also owner of The Fourth Crusade, a consulting firm specializing in automation and communications. For the last twenty years, he has worked with high availability systems in the industrial, financial, and telecommunications fields. Mr. McDaniel's company is an IBM Premier Business Partner (<http://www.fourthcrusade.com>). His areas of expertise include DB2 (various platforms), software distribution (CID, NetFinity), OS/2 Warp Server, Dynamic IP, and integration with UNIX and Windows 95/NT networks. He holds various certifications from IBM including DB2 and OS/2 Warp Server.
- **Toshimasa Shimizu** is a Senior IT Specialist with IBM Japan. He was with ITSO Austin Center for five years supporting OS/2 LAN Server, OS/2 Warp Server, and OS/2 Warp-related products. For more than twenty years, he has been a Systems Engineer for large account customer projects in Japan, providing LAN system designs, distributed systems, and networking systems.
- **Timothy Sipples** is Advisory Technical Marketing Specialist with IBM and is based in Chicago, Illinois. He specializes in software distribution issues, Java, and thin clients such as WorkSpace On-Demand. He also has extensive experience with Lotus Domino, TCP/IP, and the interoperability of OS/2 Warp with NetWare. Mr. Sipples is coauthor of the IDG/IBM Press book *IBM's Official OS/2 Warp Frequently Asked Questions*.
- **Akiko Ueno** is an IT Engineer with IBM Japan in the PC Server Group. She specializes in customer service for OS/2 Warp Server and IBM PC Servers running Windows NT. She joined IBM two years ago.



Authors (from left to right): Toshi, Michael, Peter, Akiko, and Timothy. (Uwe not pictured.)

Acknowledgments

The authors wish to thank the following companies and individuals for their valuable contributions:

IBM Corporation

<http://www.ibm.com>

- Greg Althaus, TCP/IP Applications and Kernel Development, Austin, TX
- Barry Arndt, LAN Transport Development, Austin, TX
- Mel Bryant, OS/2 Warp Server Customer Focal Point, Austin, TX
- Esther Burwell, Dynamic IP and Management Development, Raleigh, NC
- Charlotte Davis, Dynamic IP and Management Development, Raleigh, NC
- Bruce Faulkner, Dynamic IP and Management Development, Raleigh, NC
- Steven French, OS/2 Server for e-business, Austin, TX
- Pratik Gupta, Dynamic IP and Management Development, Raleigh, NC
- Wayne Ha, DB2 Development, Toronto, Canada
- Robert Hanner, Network Computing Software, Austin, TX
- Bill Hartner, OS/2 Warp Server SMP Development, Austin, TX
- Juliana Hsu, DB2 Development, Toronto, Canada

- Steven King, OS/2 Warp Server Product Manager, Austin, TX
- Saravana Kumar, OS/2 Support, India
- Rachele Powell, Dynamic IP and Management Development, Raleigh, NC
- Pankaj Sinha, OS/2 Support, India

IBM's International Technical Support Organization (ITSO)

<http://www.redbooks.ibm.com>

- Marcela Adan, ITSO Rochester, Minnesota
We thank you for your contributions to this redbook.
- Martin Murhammer, ITSO Raleigh, North Carolina
We thank you for your ideas, expertise, and contributions to this redbook.
- Elizabeth Barnes, Marcus Brewer, and Tara Campbell, ITSO Austin, Texas
Thank you so much for performing the editorial work on this book.
- Steve Gardner, Jasenn McNair, ITSO Austin, Texas
Thanks for your IT support, which made the project possible.

Hewlett-Packard Corporation

<http://www.hp.com>

- Wylie McDonald, Austin
 - Cyndi Watson, Austin
- We are grateful for Hewlett-Packard's loan of a LaserJet 4000 printer with JetDirect.

Network Associates

<http://www.nai.com>

- Robert S. Kusters, Product Manager
 - Paul Farr, Director of Product Marketing
- Network Associates supplied our team with Sniffer Basic (formerly known as NetXRay), a network protocol analysis utility.

Network TeleSystems (NTS)

<http://www.nts.com>

- Bob Baumann, Dallas
 - Russ Young, Austin
- We appreciate NTS's extra support for Shadow IPserver in our labs.

Sun Microsystems

<http://www.sun.com>

- Richard Taylor, Austin
- Bill Rappy, Austin
- Glenn Fawcett, Beaverton, Oregon

Sun supplied us with an UltraSPARC workstation running SunOS Release 5.6, and we thank Sun for their technical assistance.

Comments Welcome

Your comments are important to us! We want our books to be as helpful as possible. Please send us your suggestions, criticisms, or even praise for any IBM ITSO publication using one of the following methods:

- Fax the evaluation form found in "ITSO Redbook Evaluation" on page 471 to the fax number shown on the form.
- Use the electronic evaluation form found on our Web site:

for Internet users <http://www.redbooks.ibm.com>

for IBM Intranet users <http://w3.itso.ibm.com>

- Send us an e-mail at the following Internet address:

redbook@us.ibm.com

or, if you are an IBM Lotus Notes user, you may use the following address:

Redbook Feedback/Cary/IBM@IBMUS

We cannot promise a response to each note, but we do promise to review all your comments.

Redbooks Online

The "Redbooks Online!" page, is *the* source for finding complete redbooks on the World Wide Web. You can view the complete contents of our books. The books are arranged within the structure of the IBM Redbooks CD-ROM libraries. Clearly, we would like to see a CD-ROM or hardcopy book as part of your daily reference material. These online books help you understand what is on our CD-ROMs, what is available on hardcopy, and give you immediate access to the latest books that are not yet distributed on plastic or paper media. Redbooks Online is at the following Redbooks Web sites:

For Internet users <http://www.redbooks.ibm.com>

For IBM intranet users <http://w3.itso.ibm.com>

Chapter 1. TCP/IP Basics

Many excellent publications describe TCP/IP and the Internet in a comprehensive fashion. (References to several of these publications, including Requests for Comments (RFCs), are included at the end of this chapter.) This chapter provides a short introduction to TCP/IP; so you can become acquainted (or reacquainted) with TCP/IP addressing. Familiarity with these concepts will help you master subsequent chapters.

After reading this chapter, you should understand what IP addresses, subnets, routers, and name servers are. You should also learn what the differences are between Class A, B, C, and D addresses. Static versus dynamic addressing will be discussed, along with basic information on so-called legacy protocols, such as NetBEUI.

Feel free to skip this chapter if you are already familiar with basic TCP/IP concepts.

1.1 Network Protocols

Computer networks simply deliver bits of information from one point to another. One requirement for transmitting such information is that the computer systems on each end both speak the same language, or *protocol*.

You're already familiar with many protocols and how they're defined. For example, to address a regular letter or package in the United States, you need to write the destination address on the front of the envelope. That address could look like:

Dr. John Taylor
Sickville Eye and Foot Care Center
57 Pediatric Lane, Suite 300
Sickville, VT 05400

The protocol for U.S. mail requires a specific location where you write this destination address (the front of the envelope), the name of the recipient, the business name (if any), the street address (perhaps with a suite or apartment number), the city, state, and a postal (zip) code. Additional requirements may apply, such as the amount of postage, a return address, proper packaging, and so on. All these requirements define the U.S. Postal Service protocol.

Computer network protocols require similar information in a precise format. A package of information sent over the network is called a *datagram*.

Datagrams usually include at least a destination address, source address (where it came from), length (size of package), error detection information (such as a checksum), and package contents (the bits and bytes of information being carried).

Transmission Control Protocol/Internet Protocol (TCP/IP) is one of the most popular families of network protocols, and it happens to be the one used as the basis for the Internet.

Many people think of TCP/IP in terms of layers or levels of functions. At the lowest layer, the network interface (such as a LAN) carries the network traffic over wires or other connections. The highest layer, the application layer (such as a Web browser), uses the various TCP/IP services to communicate. In between are two additional layers: the transport and internetwork layers.

The transport layer facilitates communication between applications, whether they are on the same or different systems. The main transport layer protocol is called *TCP*, and it can determine whether a message has been received or not at the other end of the connection. An alternative is *UDP* (User Datagram Protocol), which simply sends messages without checking to see whether the system at the other end has received each part. Applications that need maximum performance and that verify delivery themselves often use UDP.

Internet Protocol (IP) is one of the internetwork layer protocols, and it is responsible for properly routing datagrams to other computers across the network. (Other internet layer protocols include ICMP, IGMP, ARP, and RARP, discussed in “Other TCP/IP Terms” on page 12.) IP depends on several important addresses in order to keep track of where messages should be delivered.

1.2 IP Addresses

IP uses addresses to specify both the source and destination systems on a TCP/IP network. Each address consists of 32 bits, usually broken into four decimal numbers separated by dots (.). Each decimal number represents an 8-bit byte (an octet) in the address. For example:

00001001	01000011	00100110	00000001	32-bit address
9	.	67	.	38 . 1 decimal address

Each address can also be separated into two logical parts:

- Network Address

The network address is a lot like a postal code, because it identifies which region (or section) of the total network contains the system.

- System (or Machine, or Host) Address

The system address is similar to an apartment or suite number, because it specifically identifies a particular system within that region.

As shown in Figure 1, IP addresses belong to one of four classes depending on how the entire 32-bit address is split (a fifth class, class E, is not commonly used).

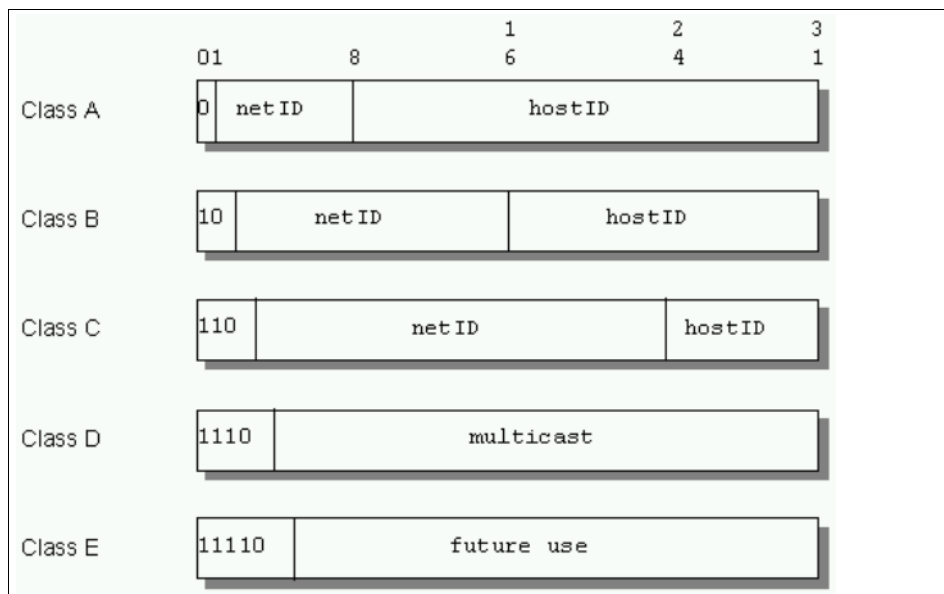


Figure 1. IP Classes

Class A Class A addresses use 7 bits for the network address portion and 24 bits for the host address. With Class A addresses, there are 126 (2^7-2) possible networks (regions) with 16,777,214 ($2^{24}-2$) possible hosts in each, a total of over 2 billion addresses. (One bit is used to identify the address as Class A, to distinguish it from other classes.)

For example, the following Class A address can be broken apart into its network and host addresses:

```

00001001 01000011 00100110 00000001
  9      .   67      .   38      .   1
^***** ++++++++ ++++++++ ++++++++

```

The first bit (marked with ^), a zero, identifies this IP address as Class A. The next 7 bits (*) provide the network number (9). The remaining bits (+) identify the specific host within network 9, in this case $67 \cdot (2^{16}) + 38 \cdot (2^8) + 1$, or 4,400,641. In other words, this IP address identifies the 4,400,641th system in the 9th network region.

Class B Class B addresses use 14 bits for the network portion and 16 bits for the host portion. These addresses provide an additional 16,382 ($2^{14}-2$) networks with 65,534 ($2^{16}-2$) hosts each, a total of over one billion additional addresses. The first two bits of a Class B address are 1 and 0.

Class C Class C addresses use 21 bits for the network part and 8 bits for the machine part, providing 2,097,150 ($2^{21}-2$) networks with 254 (2^8-2) hosts each, a total of over half a billion addresses. Class C addresses begin with 110.

As you can see, with classes A through C and 32-bit addresses, TCP/IP can provide service for a theoretical maximum of approximately 3.5 billion different hosts.

Class D Class D addresses are reserved for multicasting, a limited form of broadcasting only to other hosts sharing the same Class D address. Class D addresses begin with bits 1110.

Class E Class E addresses (beginning with 11110) are not widely used at this point in time and are reserved for future use.

In addition, some special addresses are reserved and cannot be assigned to actual systems on the network. These special addresses include:

All bits 0 Means *this*. For example, if the network address part is set to 0, the host address refers to a system on *this* (its own) network. When making initial contact on the network, a system may use this method if it doesn't know the network address. Other systems will reply with the proper network address filled in, and this proper network address can be recorded for future use.

All bits 1 Means *all*. For example, if the host address is set to all ones, the IP address identifies all systems within that

particular network region. In other words, a Class B address of 128.2.255.255 refers to all systems on network 128.2. Such an address is also called a *directed broadcast* address, because it contains a valid network address and a broadcast (all ones) host address.

Loopback

The Class A network 127 (including addresses, such as 127.0.0.1) is defined as the *loopback* network. Systems will automatically route traffic destined for these addresses back into the same system without ever communicating across the real network. Loopback addresses are often used for testing new software, to separate network problems from simple programming errors.

Private addresses

Several addresses have been reserved for private networks which are not directly connected to the Internet. These addresses include the Class A group of addresses in network 10, the 16 Class B groups of addresses in networks 172.16 through 172.31, and the 256 Class C groups of addresses in networks 192.168.0 through 192.168.255. Web servers, FTP sites, and other systems available to the public on the Internet will never have addresses beginning with 10, 172.16 through 172.31, or 192.168. These addresses are quite useful for testing purposes, or for totally private use, and you can use them without having to contact someone to reserve real addresses.

You can determine whether an address is Class A, B, C, D, or E by simply examining the first octet, as shown in Table 1.

For an address of $x.0.0.0$, if x is...

Table 1. Class Determination Table

...from (lowest)...	...to (highest), then...	Class
00000000 0	01111111 127	A
10000000 128	10111111 191	B
11000000 192	11011111 223	C

...from (lowest)...	...to (highest), then...	Class
11100000 224	11101111 239	D
11110000 240	11111111 255	E

Using Table 1 on page 5, and taking into consideration that 127.0.0.0 is defined as the loopback network, Table 2 summarizes the properties of Class A, B, and C addresses.

Table 2. Class Properties

	Class A	Class B	Class C
Lowest Network ID	1.0.0.0	128.0.0.0	192.0.0.0
Highest Network ID	126.0.0.0	191.255.0.0	223.255.255.0
Number of Networks	125	16,382	2,097,150
Number of Hosts per Network	16,777,214	65,534	254

1.3 IP Subnets

Suppose your company, MegaHuge Industries, is assigned a Class A address for all its systems around the world. Therefore, you have 16,777,214 possible IP addresses available to assign to all your systems. However, if you have many different buildings scattered across the globe, it can be hard to manage such a large number of addresses. It might be much easier to have individuals in each location (or even within a particular department) manage their own smaller sets of addresses. In addition, it's often bad for network performance to have broadcast traffic throughout the Class A network. To cut down on network congestion, your company may wish to divide this huge range of addresses into more manageable chunks.

Subnets were introduced to help solve these problems. By sacrificing the total number of addresses (16,777,214 for a Class A), you can divide your network into separately managed subnets (smaller groups of addresses). Class A, B, and even C addresses can be divided into subnets. Then, address assignments within a subnet can be performed locally, without having to contact a central authority to obtain additional addresses. The whole network (with subnets) still appears to be one IP network to the outside world.

For example, consider the Class A address 9.67.38.1. The network address is 9, and the host address is 67.38.1. Subnets simply extend this basic address by treating part of the host address as a subnetwork address. IP addresses then consist of four parts: the class identifier (0 for Class A in this example), the network address (9), the subnetwork address (for example, bits 8 to 25), and the host address (for example, the remaining bits 26 to 31).

A bit mask, known as the *subnet mask*, is used to identify which bits are part of the subnet address and which are still part of the host address. This 32-bit subnet mask has the bits for the host address set to 0 and all the other bits set to 1. (By convention, the subnet mask never has a one (1) following a zero (0). Both the network address and the subnet address are masked by ones. Technically, however, only the subnet number needs to be masked.)

So, for 9.67.38.1, with only bits 26 to 31 representing the host address, the subnet mask would be:

```
11111111 11111111 11111111 11000000
```

or 255.255.255.192 in decimal format. To extract the *subnet base address* (the network address and the subnet address together), a logical AND is performed. (If both bits in a particular bit position, for both the IP address and the subnet mask, are set to 1, then the result is also 1. Otherwise, the value is 0.) In this case:

```
00001001 01000011 00100110 00000001 = 9.67.38.1      (Class A address)
11111111 11111111 11111111 11----- = 255.255.255.192 (subnet mask)
=====
00001001 01000011 00100110 00----- = 9.67.38         (subnet base address)
```

and the remainder is:

```
----- --000001 = 1      (host address)
```

Of course the subnet number (by itself) is:

```
----- 01000011 00100110 00----- = 68760
```

Any subnet number can be chosen with the exception of all zeros (this subnet) or all ones (all subnets).

One disadvantage to using subnets is that the total number of possible IP addresses available to you decreases. For example, if you have a Class C network address of 220.23.5, you have 8 remaining bits you control. Without subnets (or, more precisely, as one subnet), you have 254 possible host addresses. With two subnets, you sacrifice two bits (subnets 01 and 10; remember, all ones and all zeroes are reserved), leaving 6 bits remaining.

Therefore, you're left with two subnets with up to 62 host addresses in each for a total of 124 host addresses, less than half the previous number.

Bearing in mind both the advantages and disadvantages of using subnets, you can easily determine the number of available subnets and hosts in each subnet using Table 3 on page 8. By subnetting your network, you can more easily mix different network technologies (such as Ethernet and token-ring), overcome limitations to the number of hosts per segment, and minimize network congestion by reducing broadcast traffic.

If you have... ...then your subnet mask is...

Class A 255.X.0.0

Class B 255.255.X.0

Class C 255.255.255.X

...where your choice of *x* yields...

Table 3. Number of Subnets and Hosts per Subnet (Partial)

x	Number of Required Subnets	Number of Possible Hosts in Each Subnet		
		Class A	Class B	Class C
0	1	16,277,214	65,534	254
128	invalid	invalid	invalid	invalid
192	2	4,194,302	16,382	62
224	6	2,097,150	8,190	30
240	14	1,048,574	4,094	14
248	30	524,286	2,046	6
252	62	262,142	1,022	2
254	126	131,070	510	invalid
255	254	65,534	254	invalid
...	invalid

Note: A Class C network cannot contain more than 62 subnets. Also, Table 3 only shows the possible number of subnets and hosts per subnet when only one octet is used for subnetting. For Class A and Class B networks, additional bits can be used for subnetting, as suggested by the last row in the table.

Supernetting, the opposite of subnetting, treats multiple networks as if they belong to one larger network. For example, a company that may need up to 2,032 possible hosts could be assigned 8 Class C network IDs ($8 \times 254 = 2,032$) to form one supernet. For example, if those 8 Class C networks are 220.78.168 through 220.78.175, then the subnet mask (to create the supernet) would be 255.255.248.0, which corresponds to the subnet mask used for a Class B network with 30 subnets of 2,046 possible hosts in each. Supernets are formed primarily to simplify routing on a TCP/IP network.

1.4 IP Routing

As mentioned above, one reason for dividing a network into subnets is to cut down on the amount of broadcast traffic throughout the network. Generally, a system on one subnet can send IP datagrams to a system on another subnet only by working through an intermediary, called an *IP router* or *gateway*.

Again, let's use the Postal Service as an example. A mail delivery system without subnets would mean that every delivery truck visits every home and business until all the packages and letters are delivered. Such a system might work, and pickup service would be quite frequent, but delivery would take a long time. Consequently, the Postal Service uses subnets, in essence, with one truck serving each particular neighborhood. Packages and letters picked up by that truck are routed to one or more central handling facilities, then further routed to other trucks on the delivery side. However, if your package or letter is destined for your neighbor's house (on the same subnet), there's very little routing involved, and the same truck will deliver that particular mail. The size of each truck route, the number of central facilities, the speed of transfers, and so on, determine the overall level of service and performance.

Similarly, there are two types of IP routing. *Direct routing* (or *direct delivery*) means that an IP datagram can be sent directly to another system which is on the same subnet, without involving an intermediate gateway. *Indirect routing* occurs when the destination host is not on a network directly attached to the source machine. One or more gateways must handle the traffic, and the IP address of the first gateway must be provided to the machine trying to send the datagram. This address is called the *gateway address* or *router address*. Each system on the network maintains a *routing table* to help determine which type of routing to use, indirect or direct, to reach another system. Three types of routings can be found in the table:

1. direct routes
2. indirect routes

3. default routes (in case the destination IP network is not found elsewhere in the table)

A routing table might look like this:

destination	router	interface
129.7.0.0	129.7.0.1	ethernet
128.15.0.0	128.15.0.1	modem
128.10.0.0	128.10.0.5	Token-Ring
default	128.10.0.5	Token-Ring
127.0.0.1	127.0.0.1	loopback

1.5 Assigning IP Addresses

In short, there are generally three pieces of information a system needs in order to start communicating on a TCP/IP network: an IP address (to uniquely identify that system on the network), a subnet mask (to help divide that 32-bit address and determine the subnet and network parts), and at least one default router address. (If a machine does not need to communicate beyond its immediate subnet, the router address and the machine's own IP address are set the same.) These three values represent the bare minimum needed for a system to participate in the TCP/IP world, and they are vital. Yet, with networks changing so quickly, manually programming these values into each and every device attached to the network (and reprogramming them as they change) can quickly become tiresome.

IP addresses assigned to systems manually (by changing a setting at each individual system) are called *static addresses*. BOOTP and DHCP can be used to assign *dynamic addresses*.

BOOTP, the *BOOTstrap Protocol*, was one of the first attempts to automate delivery of these critical values. A client system (such as a PC) can boot up and obtain the address information needed to connect to the network from a BOOTP server, where all the information is kept and managed centrally.

Although BOOTP is still widely used, its successor, *DHCP* (*Dynamic Host Configuration Protocol*), provides much more flexibility. IP addresses can be used and reused according to rules set in the DHCP server, without having to make modifications for every change in the network. Also, many DHCP servers can still provide address information to older BOOTP clients. A well designed DHCP server can save a lot of work and help get your TCP/IP network up and running more quickly. If you'd like to start exploring DHCP, read Chapter 2, "Up and Running with DHCP" on page 17.

1.6 Name Servers

While 32-bit numbers are easy for computers to understand, human beings tend to prefer names. So, each system with an IP address can be assigned one (or more) alphanumeric name(s). For instance, 9.67.38.1 could be called charlie, and 9.67.38.2 might be alice. Subnets and network addresses can also be assigned names, such as com, ibm.net, or kingscollege.ac.uk. Names assigned to subnets, networks, or any arbitrary collection of IP addresses are called *domain names*. Therefore, a system's *fully qualified name* might be charlie.ibm2.ibm.com or alice.kingscollege.ac.uk. Domains are designed to save typing and to provide some structure to the naming of systems. Generally, you do not have to include the domain name if you're simply trying to reach another system within the same domain.

The system that has the job of keeping track of which names correspond to which IP addresses is called a *domain name system* (DNS) *server*. A DNS server simply contains a table of addresses with their corresponding names. Lookups can be performed in either direction, but usually, a DNS server translates names into numeric addresses on behalf of clients. For any machine to take advantage of IP names, it must know the numeric address of at least one name server. This address is the fourth critical piece of information most machines need in order to get basic TCP/IP service, although it's optional, since it's still possible (albeit cumbersome) to reach other systems using nothing but numeric addresses.

Most systems can accept up to three name server addresses (a primary and up to two backups), contacting each in turn in order to look up the IP address that corresponds to a particular name. Yet, there are thousands of DNS servers connected to the Internet, each handling a small portion of the vast list of names and addresses. To handle these lookups most efficiently, one DNS server can forward a lookup request it cannot satisfy to another DNS server. Often lookup responsibilities are divided according to the dots (.) in the fully qualified name. For example, charlie may be given one name server address, and that local name server (DNS 1) knows the names and addresses of all the systems in the ibm2.ibm.com group. When charlie requests an IP address for gadget.att.com from DNS 1, DNS 1 may forward the request to DNS 2 (serving ibm.com), which then forwards the request to DNS 3 (serving com), which then forwards the request to DNS 4 (serving att.com), which then answers with the proper IP address for gadget.att.com. In practice, name servers record many of the most recent lookups, to help cut down on network traffic. For example, if samuel.ibm2.ibm.com requests the IP address for gadget.att.com from DNS 1 just after charlie does, DNS 1 may be able to provide the answer without bothering DNS 2, DNS 3, or DNS 4.

Note that the capitalization of a TCP/IP name does not matter. For example, `charlie.ibm.us2.ibm.com`, `CHARLIE.IBM.US2.IBM.COM`, and `ChArLiE.ibm.us2.IBM.cOm` will all be treated by a DNS the same way, and the DNS will return an IP address of 9.67.38.1 for each of these variations. It's also quite common to have multiple listings, known as *alias names*, with several different names corresponding to the same numeric IP address. Many companies try to gain a marketing advantage by registering multiple names, hoping that someone will stumble into their Web site by typing one of several generic names such as `www.casino.com` or `www.gambling.com`.

A *static DNS* requires someone (a person) to manually edit and update the lookup table whenever an IP address gets assigned or reassigned to a particular name, or when a name is no longer used because the system is out of service. Needless to say, this task can be tedious. Chapter 4, "Serving Names" on page 87, explores how a *Dynamic DNS* (DDNS) server performs, with information on how to make communication between name servers and hosts more sophisticated.

Registration of IP addresses and domain names on the Internet is currently managed by a central administrative body called InterNIC, the Internet Network Information Center. InterNIC's Internet Web page can be found at:

<http://rs.internic.net/rs-internic.html>

1.7 Applications That Use TCP/IP

Programmers have written many applications which communicate using TCP/IP. Web browsers (HTTP), FTP (file transfer), Telnet (terminal emulation), LPR/LPD (printing over TCP/IP), REXEC and RSH (starting programs remotely over the network), POP/SMTP/IMAP (e-mail services), NFS (file sharing), X-Windows (graphics terminal emulation), NNTP (news), SNMP (network management), and Java, among other examples, can all use these common services. Although many variations exist for specific platforms (such as *Winsock* for Microsoft Windows), a *sockets interface* allows programmers to write applications that communicate with TCP/IP.

1.8 Other TCP/IP Terms

To work properly, IP protocol needs some additional help to manage traffic flow efficiently. These terms are explained briefly; please consult the appropriate references for more detailed information:

- *Internet Control Message Protocol* (ICMP), a part of IP, helps report errors in datagram delivery. ICMP can also help discover routers and *maximum*

transmission units (MTUs) (see below) along the path the datagram travels. PING, the popular TCP/IP application used to check the connection between two systems on the network, uses ICMP. RFC 792 describes ICMP in detail.

- The maximum transmission unit (MTU) is the size of the IP datagram, which can be adjusted depending on network conditions. All systems on a TCP/IP network are required to handle MTUs ranging from 576 to 65,535 octets. Each datagram typically packages 20 octets of identifying information, such as the destination address, as part of the structure.
- *Internet Group Management Protocol (IGMP)* allows systems to participate in IP multicasts and to cancel such participation. Additionally, IGMP provides routers with the ability to check hosts to see if they are interested in participating in multicasts. RFCs 966, 1112, and 1458 discuss IP multicasting.
- *Address Resolution Protocol (ARP)* maps IP addresses to hardware addresses on a network. (Hardware addresses are often Ethernet or token-ring network adapter addresses consisting of 12 hexadecimal digits.) *Reverse ARP (RARP)* provides the reverse lookup. See RFC 826 for more information on ARP.
- *SLIP* (RFC 1055) and *PPP* (RFCs 1717 and 1661) provide TCP/IP services over serial lines, such as modems and ISDN (digital telephone) connections. Chapter 7, “Mobile Users” on page 279, provides much more information on PPP.
- *IPv6 (Internet Protocol version 6)* is proposed to help alleviate some of the address constraints and other shortcomings of TCP/IP as it becomes ever more popular, although an upgrade to these capabilities will take time. Fortunately, IPv6 simply extends today's TCP/IP; so the skills you develop in this book should help you prepare for these protocol enhancements as they become available. See RFCs 1883 through 1887 for more information on IPv6.
- *Firewalls* help protect an *intranet* (private TCP/IP network) from unwanted access from the Internet, such as people trying to break into servers. At the same time, people on the intranet can still get access to Internet systems. Firewalls vary in capabilities, but the two basic types are *proxy* and *SOCKS* firewalls. Firewalls can also help link two separate intranets via the Internet (and still provide security) using a technology called *Virtual Private Networks* (VPN). Chapter 8, “Security of DHCP and Dynamic DNS” on page 309, takes a look at firewalls in the context of overall TCP/IP network security.

- Traditional LAN (Local Area Network) protocols can be carried by TCP/IP, to provide file and print sharing services for PCs without contaminating a TCP/IP network with other protocols. (Many computer networks are perfectly capable of mixing protocols, but many network managers prefer to standardize on one protocol.) For example, RFC 1234 describes how Novell's IPX operates over TCP/IP; so traditional NetWare servers and clients can communicate via TCP/IP. NetBIOS over TCP/IP (used by IBM's OS/2 Warp Server and clients, IBM DOS/Windows LAN Services, Microsoft Windows 95, Windows NT, and Windows for Workgroups), and explained by RFCs 1001 and 1002, supports applications written to the NetBIOS programming interface. NetBIOS isn't actually a protocol. Like the sockets interface, it's a widely accepted way of writing network-savvy applications. The NetBIOS interface can be supported by a number of underlying network protocols, including NetBEUI (NetBIOS over IEEE 802.2), TCP/IP (TCPBEUI), and even SNA. Like TCP/IP, NetBIOS also relies on names to communicate with other systems on the network, although these names behave much differently. A *NetBIOS Name Server* (NBNS) can help provide better NetBIOS service over a TCP/IP network. Consult Chapter 5, "Integrating File and Print Services" on page 177 for details on NetBIOS over TCP/IP issues.

1.9 Related Publications

This chapter has provided just enough information to understand most of the concepts explored in this book, but there's far more technical information on TCP/IP available. For more information on TCP/IP, please refer to the following publications:

- *Internetworking with TCP/IP, Volume I, Principles, Protocols and Architecture*, third edition, Prentice-Hall, Inc., 1995, by Douglas E. Comer; ISBN 0-13-216987-8.
- *TCP/IP Tutorial and Technical Overview*, fifth edition, IBM Corp., 1995, GG24-3376-04, and Prentice-Hall, Inc., 1995, by Eamon Murphy, Steve Hayes, Matthias Enders; ISBN 0-13-460858-5.
- *IPng and the TCP/IP Protocols*, John Wiley & Sons, Inc., 1996, by Stephen A. Thomas; ISBN 0-471-13088-5.
- *Communications for Cooperating Systems - OSI, SNA and TCP/IP*, Addison-Wesley Publishing Company, Inc., 1992, by R.J. Cypser; ISBN 0-201-50775-7.
- *Request for Comments (RFCs)*

There are more than 2,200 RFCs today. For those readers who want to keep up-to-date with the latest advances in TCP/IP, the ever-increasing number of RFCs and Internet Drafts (IDs), published by the non-profit Internet Engineering Task Force, are the best sources. RFCs can be viewed on the Internet at:

<http://www.isi.edu/rfc-editor>

Chapter 2. Up and Running with DHCP

In this chapter, we show how to set up a simple TCP/IP network with one subnet using dynamic IP. Each system connected to this simple network will be provided address, subnet mask, router address, and name server information to permit the most basic connection. These critical network connection values will be provided using DHCP, and we will explain why having dynamic addressing helps your network run more smoothly. Furthermore, Chapter 3, “DHCP Server and Client Interaction” on page 75, explains in detail all interactions between DHCP clients and DHCP servers.

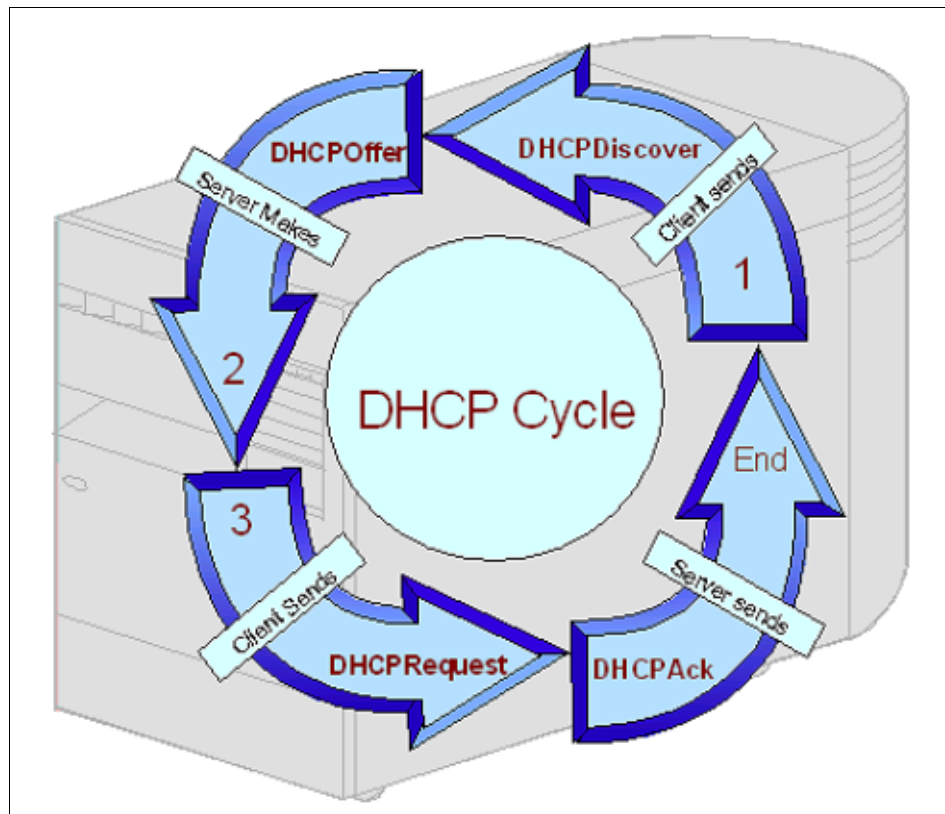


Figure 2. DHCP Cycle

Here you will discover how to connect just about any type of computing device to your DHCP network for true interoperability, as Figure 2 indicates. As the title suggests, this chapter should get you up and running quickly. The network you create in this chapter can be useful in many small environments.

In later chapters, we demonstrate how to build upon the basics introduced here.

2.1 Why DHCP?

Computer networks always seem to be changing. New devices (PCs, printers, and so on) must be attached, old devices disconnected, new branches added, mobile workers hosted, temporary employees accommodated -- all these changes can happen every day. Managing all that change can prove a major undertaking without systems that respond automatically to changing demands.

On a TCP/IP network, each system must have an IP address, subnet mask, and router address (at a minimum) in order to communicate, as discussed in Chapter 1, "TCP/IP Basics" on page 1. At least one name server address should generally be provided as well. Unless your network contains only a small number of systems, which rarely (if ever) change, managing all these address settings manually can become overwhelming. Using a DHCP server to manage these addresses for you can help make your TCP/IP network more reliable, flexible, and less time consuming.

2.2 OS/2 Warp Server as a DHCP Server

OS/2 Warp Server makes a fine DHCP server, with sophisticated management features and excellent performance, particularly with OS/2 Warp Server Advanced SMP. IBM's TCP/IP Version 4.1 for OS/2, which enhances OS/2 Warp Servers DHCP capabilities, is available from IBM Software Choice at the following Internet Web site:

<http://www.software.ibm.com/swchoice>

Note: Some features in IBM's Software Choice catalog, such as TCP/IP Version 4.1 for OS/2, are fee-based features.

2.2.1 Installation

Be sure to select OS/2 Warp Server's DHCP server function during installation. We also recommend making the following additions to OS/2 Warp Server in this order:

- FixPak 36 (or later)
- Netscape Navigator 2.02 (June, 1998, or later)
- Feature Installer 1.2.2 (or later)
- Java 1.1.4 (or later; we recommend Java 1.1.6)

- TCP/IP Version 4.1 (includes MPTS 5.3)
- Updates to TCP/IP 4.1, MPTS 5.3, Java, and any other OS/2 Warp Server functions you plan to use (such as OS/2 LAN Server)

These additions can be found at the following Internet Web sites:

- <http://www.software.ibm.com/swchoice>
Netscape Navigator, Feature Installer, Java, TCP/IP 4.1, and other enhancements.
- <http://ps.software.ibm.com>
Updates and fixes for all IBM PC software products.
- <ftp://ftp.hursley.ibm.com>
Updates to Java.

2.2.2 DHCP Server Preparation

The first thing you need to do after the DHCP server is fully installed is to set a password for the administrator. You may also wish to make some changes to various TCP/IP settings.

Tip

TCP/IP Version 4.1 for OS/2 uses Java-based graphical setup panels. Consequently, you can manage OS/2 Warp Server's DHCP server from just about any Web browser. (See Chapter 11, "Managing Your Network" on page 369 for details.) However, when you make changes using the Java graphical panels you should remember the following:

- *Scroll bars.* Sometimes all the information cannot fit within each window. You may have to scroll up (or down) to see everything.
- *Horizontal adjustments.* You may be able to read some information better if you make horizontal adjustments to some of the header windows (for example, Option Name).

We suggest that you create backup copies of the following files before starting:

```
C:\CONFIG.SYS
C:\MPIN\BIN\SETUP.COM
C:\MPIN\BIN\MPSTART.COM
C:\TCP\BIN\TCPSTART.COM
```

where the C: drive may be different, depending on how you installed OS/2 Warp Server.

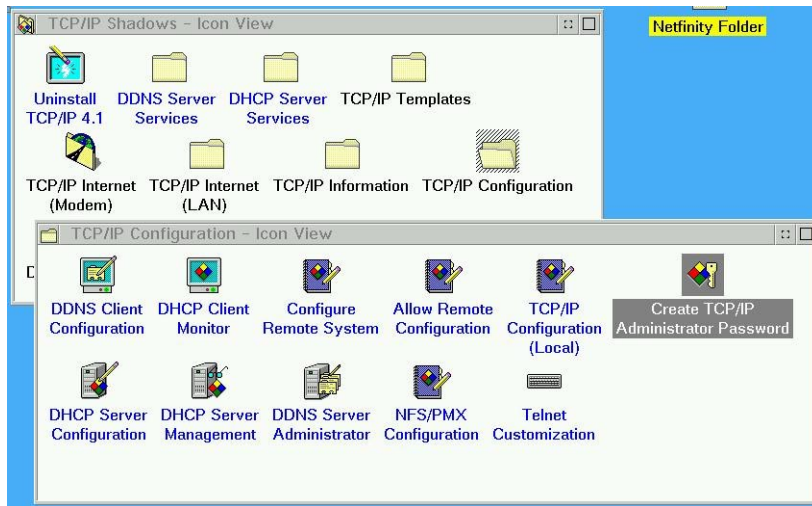


Figure 3. [Warp Server] TCP/IP Version 4.1 Configuration Folders

To change the TCP/IP administrator password:

1. Open your **TCP/IP Shadows** folder and then the **TCP/IP Configuration** folder (Figure 3).
2. Double-click on **Create TCP/IP Administrator Password**. Enter your administrator password twice, and click on **OK** (Figure 4). Remember this password!

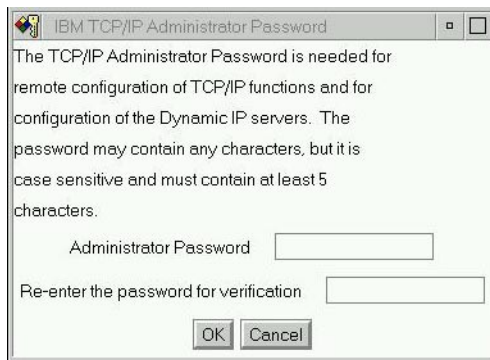


Figure 4. [Warp Server] TCP/IP Administrator Password Window

Then, you can perform some basic setup using the TCP/IP configuration notebook. (Some of these settings may already be filled in depending on whether you entered any TCP/IP settings during installation of OS/2 Warp Server.)

3. In the **TCP/IP Configuration** folder, double-click on **TCP/IP Configuration (Local)**. This starts the TCP/IP Configuration Notebook. (Figure 5 shows the values we used.)
4. Assign a static IP address to your DHCP server. Click on the **Network** tab and, with **LAN interface 0** highlighted, enter the IP address and subnet mask for the server. (The first network adapter in your server running TCP/IP is normally assigned LAN interface 0.) Click on the **Manually Using** button, if necessary, and make sure that the **Enable Interface** checkbox is checked.

The server's IP address and subnet mask are part of the Basic choices. None of the advanced options are needed at this time.

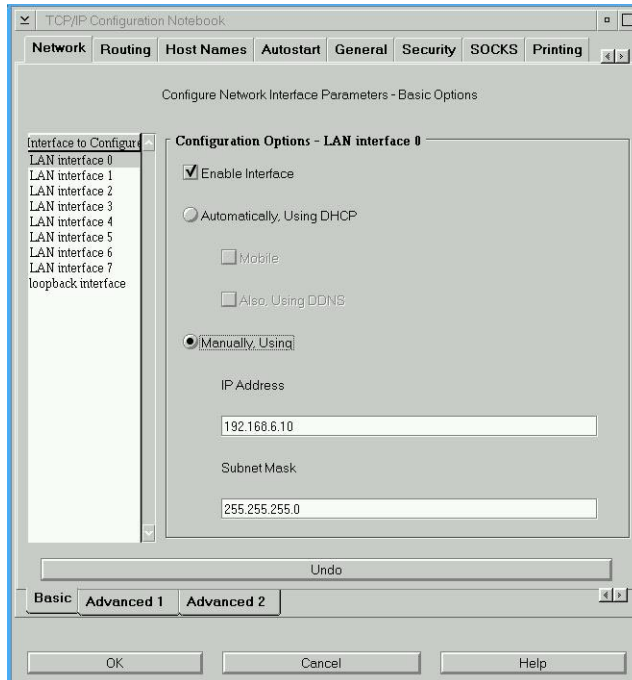


Figure 5. [Warp Server] TCP/IP Configuration Notebook

5. Click on the **Routing** tab and then **Add**. We entered our server's own IP address as the **default Router Address** for our one subnet network. You may already have a default route defined, which you can change if needed.

6. Click on the **Host Names** tab and enter your computer's hostname and domain name. For example, the hostname could be merlot and the domain name armonk.cooking.net.
7. Now, click on the **Hosts** tab at the bottom, and then **Add** a loopback address (127.0.0.1) using the hostname localhost.

We chose not to **Autostart** any services, such as FTPD or TELNETD, at this time.

8. Click on the **General** tab and choose your correct time zone. We did not use **Security**, because we are not presently providing any of the services on this page. Also, we are not using **SOCKS** or **Printing**.
9. When you are satisfied with all your entries, click on the **OK** button at bottom of the window and **OK** again to continue. Your new configuration will be saved in CONFIG.SYS and \MPTN\BIN\SETUP.COM. Shut down and reboot your server now so that these changes take effect.

You have now completed basic setup of TCP/IP 4.1 for your DHCP server.

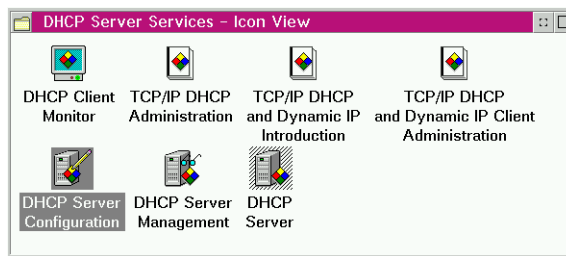


Figure 6. [Warp Server] DHCP Server Services

2.2.3 Basic Configuration

After completing basic TCP/IP setup, as shown in the previous section, you are now ready to create a simple DHCP configuration.

1. Open up the **TCP/IP Shadows** folder again, then **DHCP Server Services**, and double-click on **DHCP Server Configuration** (Figure 6). A DHCP Server Configuration window should appear to request your TCP/IP administrator password (Figure 7 on page 23).



Figure 7. [Warp Server] Password for DHCP Server Configuration

2. Enter your password and press **Enter** (or click on **OK**). Be patient, after you press **Enter**, it will take some seconds before the **DHCP Server Configuration** window appears (Figure 8 on page 24). You may have to click on the window to bring it to the foreground. The mouse cursor will change shape as you move it around the window, indicating the areas which can be enlarged or reduced in size.

We recommend that you use this DHCP server configuration window to make any configuration changes, rather than editing DHCP configuration files directly. This Java program automatically checks your entries as they are made. If you edit the configuration file directly, it can become very difficult to resolve any errors.

A default configuration file should be loaded when you first open the DHCP Server Configuration window, but you may wish to create a new file. (The configuration file is `\MPIN\ETC\DHCPD.CFG`. You should keep backups of this critical file.) You can also load a sample file by selecting **File**, then **Load Sample**.

The remaining steps describe what we did to create a DHCP configuration for our one subnet network.

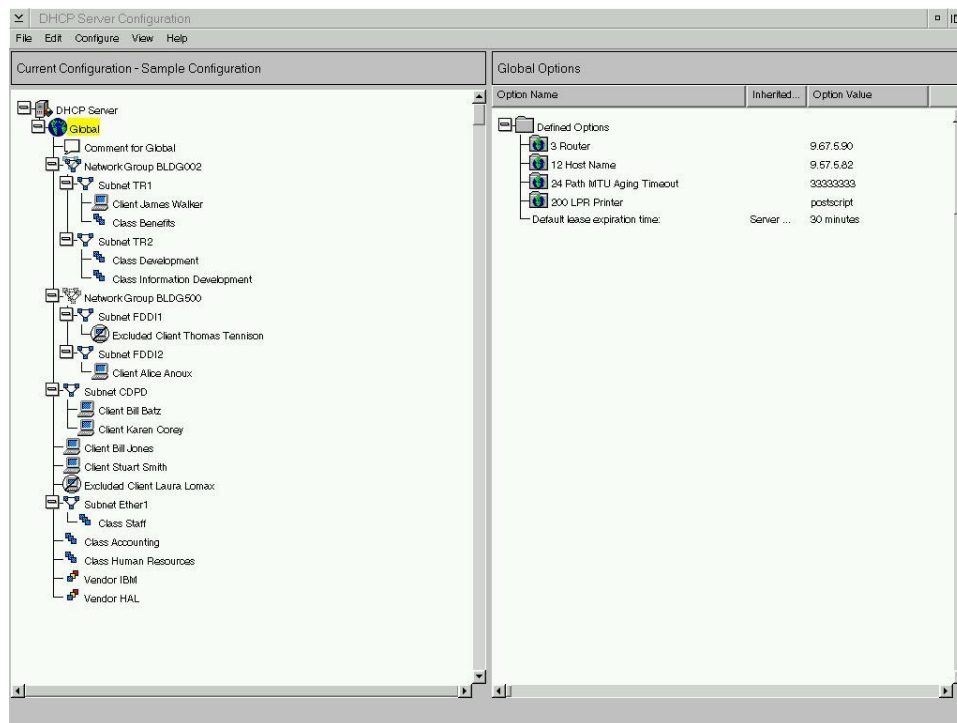


Figure 8. [Warp Server] DHCP Server Sample Configuration

3. Click on **File** and then **New** (see Figure 9 on page 25).

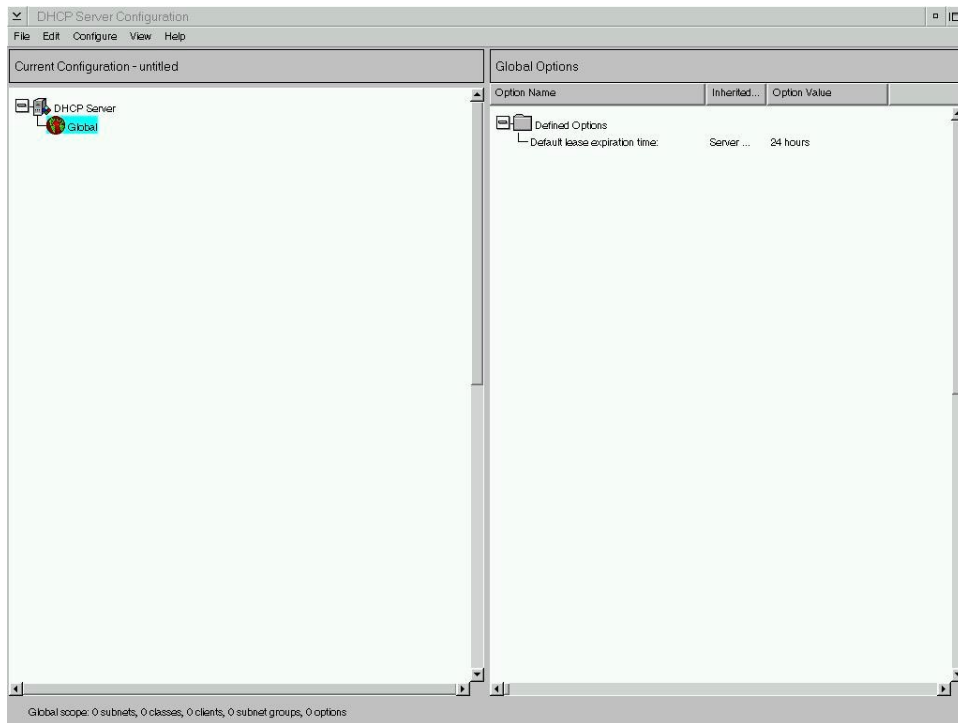


Figure 9. [Warp Server] DHCP Server New Configuration

4. Click on **Configure**, then **Add Subnet**. After a few seconds, a notebook should open. Fill in the values as desired.
5. Click on the **DHCP Options** tab. As shown in Figure 10 on page 26, we chose only option 1, the subnet mask (255.255.255.0), and option 15, the domain name (armonk.cooking.net).

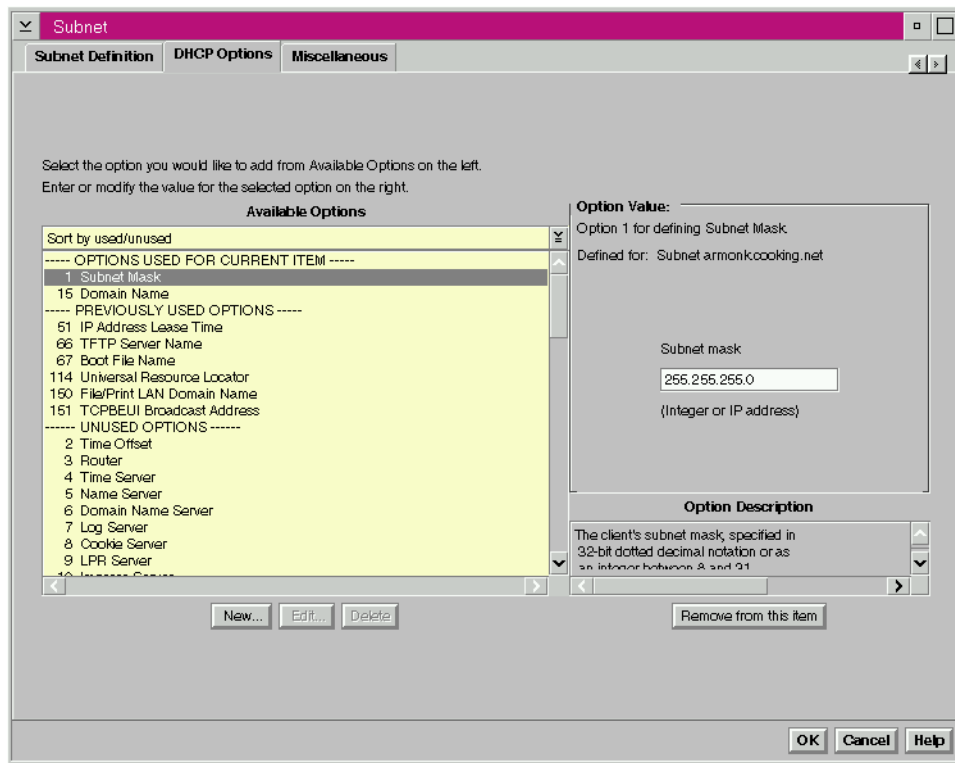
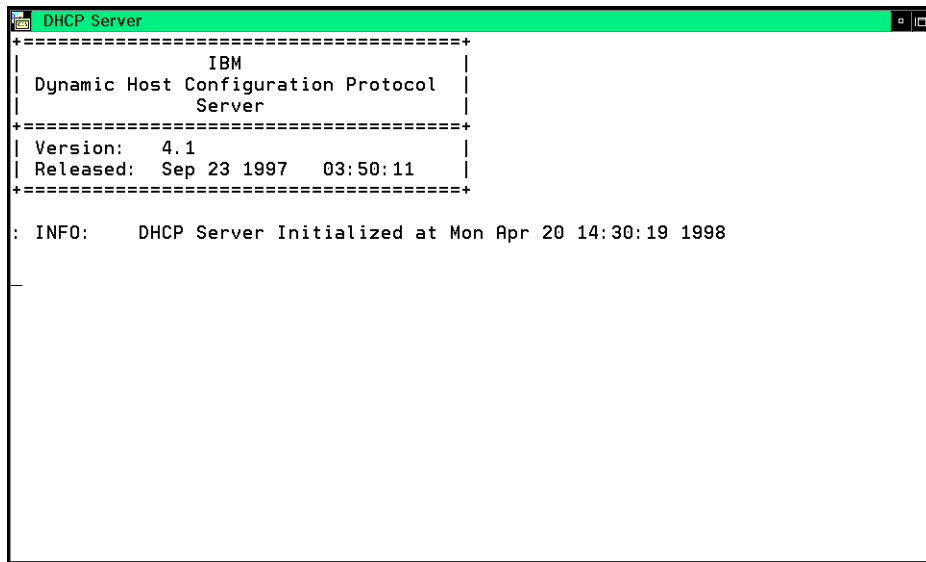


Figure 10. [Warp Server] DHCP Subnet Notebook

6. Save your changes [**File — Save**].
7. Start the DHCP server to check your new configuration. Double-click on the **DHCP Server** icon in the DHCP Server Services folder. As shown in Figure 11 on page 27, you should see a message when the server starts (in a separate window) that says *Server Initialized*.



```
=====+
|          IBM          |
| Dynamic Host Configuration Protocol |
|          Server      |
|-----+-----+
| Version:   4.1        |
| Released:  Sep 23 1997   03:50:11  |
|-----+-----+
: INFO:      DHCP Server Initialized at Mon Apr 20 14:30:19 1998

```

Figure 11. [Warp Server] DHCP Server Initialized

If there are problems, the DHCP.DLOG file is a good place to look. This text file usually resides in the root directory. You may change the location of this file if you wish by double-clicking on **DHCP Server** in the **DHCP Server Configuration** window. A settings notebook appears, as shown in Figure 12 on page 28. You can enter the full path to a new log file in the **Log file name** field. You can also choose which kinds of errors should be logged to this file by checking or unchecking each checkbox.

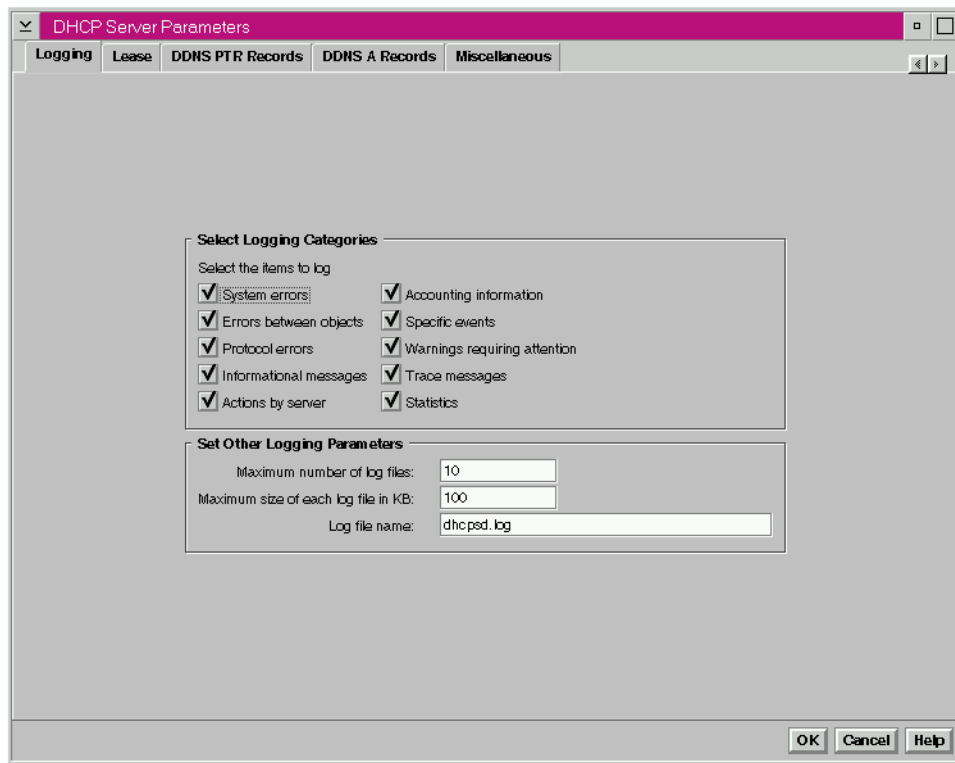


Figure 12. [Warp Server] DHCP Log File Notebook Settings Notebook

2.2.4 Testing and Verification

Now that your DHCP server is running, your DHCP clients should be able to get IP addresses and parameters from the server without difficulty. We tested many different DHCP clients with the OS/2 Warp Server DHCP server, including:

- OS/2 Warp 4
- Windows 95
- Windows NT Workstation 4.0
- Apple Macintosh
- IBM Network Station
- Linux
- DOS, Windows 3.1, and Windows for Workgroups
- Sun Microsystems Solaris
- IBM WorkSpace On-Demand
- Hewlett-Packard LaserJet 4000 Printer with JetDirect

None of these clients had any problems communicating with the IBM DHCP server. (You can learn how to configure each of these clients in 2.4, “DHCP Clients” on page 39.)

To determine which IP addresses are being used by your clients, open an OS/2 command line window and enter the following command on the server:

```
DADMIN -S | MORE
```

(You may also wish to check address assignments at each client. Or, you may wish to examine the DHCP log file on the server.)

You can also use the `DADMIN` command to perform any of the following tasks:

- Re-initialize the DHCP server. (The server re-reads its configuration file and adopts any changes.)
- Delete a lease.
- Control server tracing.
- Display client information.
- Display IP address information.
- Display server statistics.

DADMIN Command Syntax

```
dadmin [-?]
        [-v] [[-h]<host>] [-f] -d <ipaddress>
        [-v] [[-h]<host>] -i
        [-v] [[-h]<host>] -s
        [-v] [[-h]<host>] -t <on/off>
        [-v] [[-h]<host>] -n <intervals>
        [-v] [[-h]<host>] -q <ipaddress>
        [-v] [[-h]<host>] -p <ipaddress>
        [-v] [[-h]<host>] -c <client id>
        [-v] [[-h]<host>] -u <User Password>
        [-v] [[-h]<host>] -x
        [-v] [[-h]<host>] -a
```

where:

- ? Displays usage information.
- v Enables verbose mode. This mode provides additional information for debugging purposes.
- h Specifies either the IP address or the hostname of the DHCP server. If no server is specified, the local server is assumed.
- f Forces deletion of the lease without prompting.

- d Deletes the lease for the specified IP address.
- i Re-initializes the specified server.
- s Displays all IP addresses currently leased.
- t Specifies server tracing. Use a value of `ON` to start tracing or `OFF` to stop tracing.
- n Requests statistics for the specified server. The value is a decimal integer indicating the number of intervals from 0 to 100. For example, a value of 3 returns a summary record that includes totals information, the current interval record, and the 3 most recent history records. A value of 0 returns a summary record of activity since the last summary.
- q Requests the IP address information.
- p Requests the address pool information.
- c Requests information for one or more clients that match the client ID.
- u Specifies the administrator password. If the DHCP server is local, this parameter is not required. If the password contains blanks, you must enclose the password in double quotes (").
- x Specifies that the DHCP server is older than the version supplied with IBM TCP/IP Version 4.1. Access is controlled by the `RHOSTS` file on the DHCP server, which requires either a `HOSTS` file or DNS name resolution.
- a Specifies that no administrator password checking is to be performed. This parameter is used for DHCP servers that are not running on IBM TCP/IP Version 4.1 or on IBM eNetwork Network Station Manager TCP/IP for Windows NT.

2.3 Windows NT as a DHCP Server

This section describes how to set up a simple DHCP server using Microsoft Windows NT Server 4.0.

2.3.1 Installation

When you install Windows NT Server 4.0, the DHCP server is not installed by default. If you have not installed NT's DHCP Server yet, follow these steps:

1. Click on the **Start** menu. Select **Settings** and then select **Control Panel**.
2. Double-click on the **Network** icon.

3. Click on the **Services** tab in the Network notebook then click on the **Add...** button.
4. Select **Microsoft DHCP Server** from the list and then click on **OK**.
5. When prompted, enter the path to the Windows NT files (Windows NT Server CD-ROM) and click on **Continue** to copy the files and finish the installation.

2.3.2 DHCP Server Preparation

We assume a network with one subnet for this example. All clients attached to the network were previously configured using static IP addresses, including static DNS server addresses. The following changes will be made in our sample network:

- Adding a DHCP server to the network.
- Configuring all clients to obtain IP addresses, subnet mask, router address, and name server information from the DHCP server.

The following values represent our sample network configuration:

Network ID	192.168.8.0
Subnet Mask	255.255.255.0
Domain Name	bellevue.cooking.net
DNS Server	192.168.8.10
DHCP Server	192.168.8.10

You should plan your own network configuration before setting up any DHCP server. You should take the following requirements into consideration:

- Non-DHCP clients must have static IP addresses, and these must be excluded from the IP address pool.
- The DHCP server must be assigned a static IP address. It cannot be a DHCP client to itself.
- You should determine the range of IP addresses (the *scope*) that can be assigned to DHCP clients.
- Any other DHCP option values (such as name servers) should also be defined for the clients.
- The duration of the IP address lease must be chosen.

Much like 2.2, “OS/2 Warp Server as a DHCP Server” on page 18, we chose to have our DHCP server handle the private Class C addresses 192.168.8.1 through 192.168.8.254, with 192.168.8.1 through 192.168.8.10 excluded from this pool in order to reserve a small set of static addresses. One of those

reserved addresses, 192.168.8.10, was assigned to the DHCP server itself (which happened to also act as a DNS server). The subnet mask was 255.255.255.0, and the lease duration was set to one day.

We also chose the following option values for our sample network:

Option 003: Router	192.168.8.1
Option 006: DNS Server	192.168.8.10
Option 015: Domain Name	bellevue.cooking.net

Before you start configuring the DHCP server, you must make sure that TCP/IP is set up and configured correctly. In particular, the DHCP server must have a static IP address. To configure TCP/IP manually, follow these steps:

1. Open the **Control Panel** and then open the **Network** icon.
2. Click on the **Protocols** tab.
3. Select **TCP/IP Protocol** and click on **Properties**.
4. Click on the **IP Address** tab. Check **Specify an IP address** and type in the server's IP address, subnet mask, and default gateway address.
5. Click on the **DNS** tab and type in the hostname and domain name for the server.
6. Under **DNS Service Search Order**, click on **Add**.
7. In the **DNS Server**, box type in the IP address for your server (assuming it is also acting as a DNS server) and click on **Add**.
8. Click on **OK** to close the notebook.
9. Click on **OK** to close the **Network** notebook.

2.3.3 Basic Configuration

After assigning a static IP address to the DHCP server, as shown in the previous section, you are now ready to create a simple DHCP configuration.

1. By default, the Microsoft DHCP server service starts automatically when the computer is started. If the DHCP server is not running, open the **Service** icon in the **Control Panel**, select the **Microsoft DHCP Server** from the service list, and click on the **Start** button. You can also start the DHCP server from the command line, using the following command:

```
NET START DHCPSEVER
```

2. Start the DHCP manager. Click on the **Start** menu, select **Programs**, then select **Administration Tools (Common)**, and click on **DHCP Manager**.

3. You can manage both local and remote DHCP servers using this utility. If you want to access a remote DHCP server, select **Server** and then select **Add** from the menu. The **Add DHCP Server to Server List** window will appear. Enter the address(es) of the remote DHCP server(s) you wish to manage and click on **OK**.
4. Next, create the scope for the subnet. In the DHCP manager window, highlight the **Local Machine** (or the remote server you wish to manage), click on the **Scope** menu, and then select **Create**. Fill in the following information (see Figure 13 on page 34):

Start Address	Starting IP address for the scope (e.g. 192.168.8.1).
End Address	Ending IP address for the scope (e.g. 192.168.8.254).
Subnet Mask	Subnet mask assigned to the clients (e.g. 255.255.255.0).
Exclusion Range	You can exclude any static IP addresses from the scope using this option. Type in the start address and the end address, and then click on the Add button. These addresses will not be assigned to DHCP clients.
Lease Duration	Duration of the address lease. Click on the Limited To button and then type in the number of days, hours, and minutes to set the lease duration.
Name	A name for the DHCP scope. This name appears in the DHCP manager solely for your convenience.

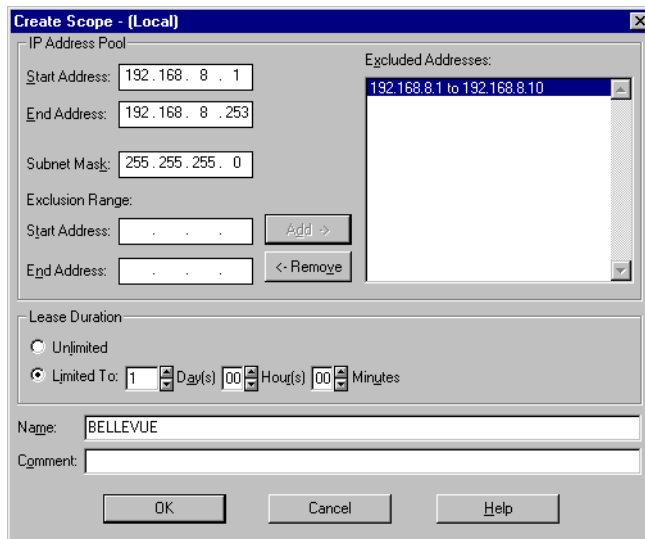


Figure 13. [Windows NT] Creating a Scope

5. After filling in all the information, click on **OK** to save the configuration. A message will appear to remind you that the scope has not been activated yet. If you click on **Yes**, you will activate the scope immediately. However, you should not activate a new scope until you have specified its DHCP option values, so click on **No**.
6. Now, you should set the option values passed to DHCP clients. In the **DHCP Servers** list, highlight the scope you just created, click on the **DHCP Options** menu, then select **Scope**.
7. In the Unused Options list, choose an option to be added and click on the **Add** button. The option you selected is moved to the Active Options list. (We configured options 3, 6, and 15, as shown in Figure 14.)

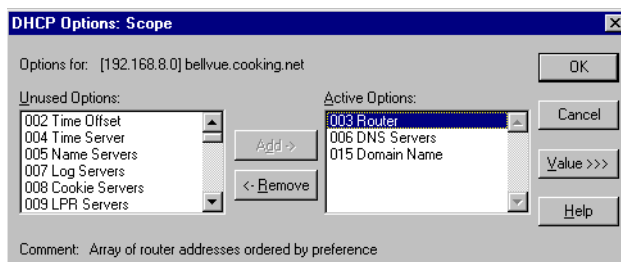


Figure 14. [Windows NT] Adding DHCP Options

8. To define a value for each option, highlight an option in the Active Options list and click on **Value**. Enter a value in the appropriate field for the option, as shown in Figure 15. Repeat for each Active Option.

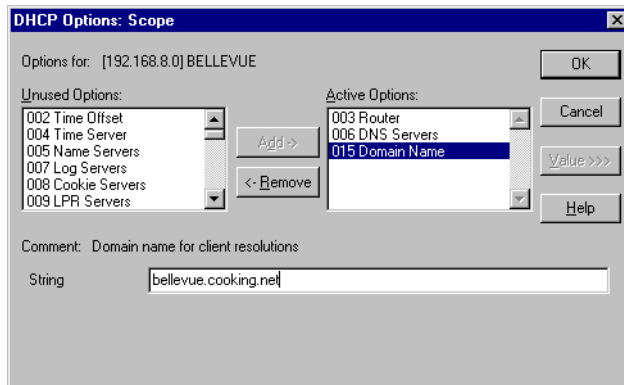


Figure 15. [Windows NT] Entering Value for Option 15

Some options can take on multiple values, such as option 6 (DNS Servers). For these options, click on **Edit Array**, enter one of the values, then click on **Add**. You can enter more values, one by one, and click on **Add** to build the full list of values for the option. When you finish building the list, click on **OK**. (See Figure 16.)

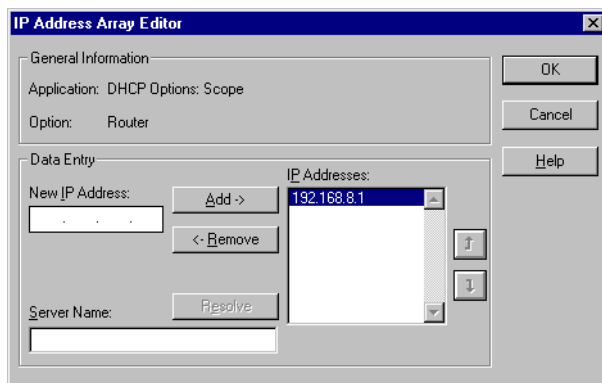


Figure 16. [Windows NT] Creating Address List for Option 3

Option 1: Subnet Mask Limitation

You cannot choose option 1 (subnet mask) from the DHCP options list in Windows NT Server. The value is taken from the scope property notebook (see Figure 13 on page 34).

When you have finished defining all the option values, click on **OK**.

9. If the scope is not active (if a small icon representing the scope in the list is grayed out), click on **Scope** and then click on **Activate**. The icon should turn yellow.

2.3.4 Testing and Verification

1. Start your DHCP clients and check whether they receive address assignments. (We tested Windows 95, Windows NT Workstation 4.0, and OS/2 Warp 4 as DHCP clients. All DHCP clients received TCP/IP configuration information through the Windows NT DHCP server properly.)
2. To check for active leases from the server, highlight the scope you want to view, select **Scope**, then select **Active Lease**. The Active Leases window (Figure 17) will appear and list active leases.

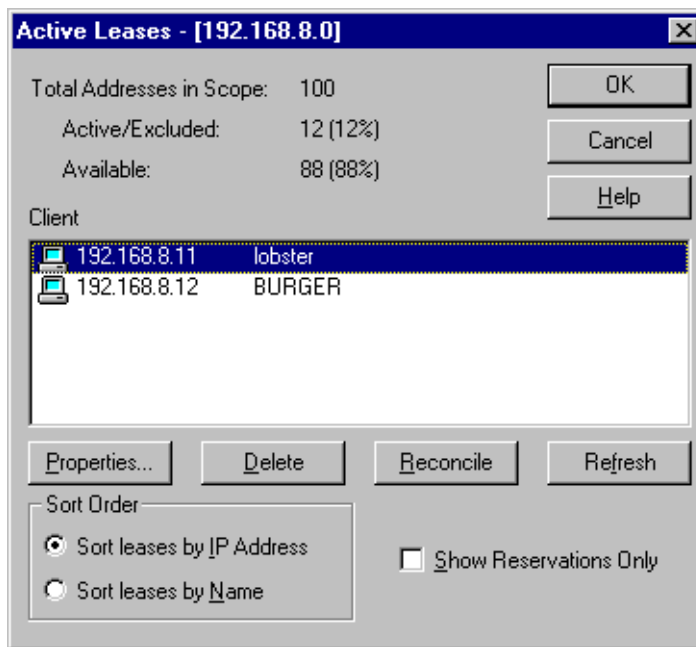


Figure 17. [Windows NT] DHCP Active Leases

Select any system in the client list, then click **Properties** to view more information about an individual DHCP client. (See Figure 18 on page 37.)

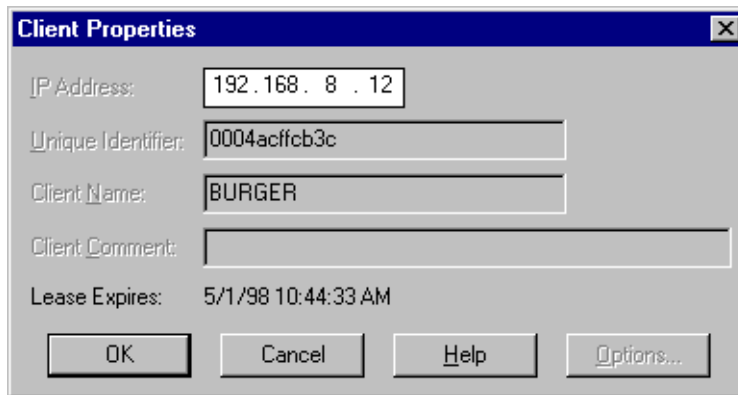


Figure 18. [Windows NT] DHCP Client Properties

2.4 DHCP Clients

Because a DHCP server, such as OS/2 Warp Server, supports open Internet standards, it can service a wide variety of clients. Most large TCP/IP networks include many different kinds of systems running a wide variety of operating systems. You may be confronted with any or all of the types of systems described in this section.

This section explains how to configure several different clients to receive TCP/IP address assignments from a DHCP server.

2.4.1 OS/2 Warp 4

OS/2 Warp 4 includes TCP/IP Version 4.0 as a standard feature. Or, you may upgrade OS/2 Warp 4's TCP/IP capabilities to Version 4.1, available from the IBM Software Choice Web site.

Download Web Sites

You can download IBM Software Choice packages, FixPaks, and other enhancements from the Internet. Please refer to Appendix B, "Where Is It? Internet and IBM Intranet Web Sites" on page 401 for more information.

Both TCP/IP versions support DHCP, but they differ in the way DHCP is activated. (Skip ahead to 2.4.1.2, "OS/2 Warp 4 DHCP Client Configuration" on page 39, if you plan to use TCP/IP 4.0.)

2.4.1.1 Installing TCP/IP 4.1

At a minimum, you must have the following software available prior to installing TCP/IP 4.1:

- OS/2 Warp 4 with TCP/IP 4.0
- Netscape Navigator 2.02 (June 1998 or later)
- Feature Installer 1.2.2 (or later)
Feature Installer provides plug-ins for Netscape Navigator that are used to install Java Version 1.1.4 and TCP/IP Version 4.1.
- Java Version 1.1.4 (or later; we recommend installing Java Version 1.1.6)
Only the Java runtime is required.

To install TCP/IP Version 4.1 on OS/2 Warp 4, follow these steps:

1. Install OS/2 Warp 4 with its built-in TCP/IP 4.0 client services (and other services as needed).
2. Install Netscape Navigator 2.02 (June, 1998, or later) for OS/2 Warp.
3. Install FixPak 6 (or later) for OS/2 Warp 4. Make sure you shutdown and reboot after installing the fixpack.
4. Install any other FixPaks (for example, File/Print Client) as needed.
5. Install Feature Installer 1.2.2 (or later). Shutdown and reboot.
6. Install Java 1.1.4 (or later; we recommend installing Java 1.1.6). Only the runtime is required, but you may wish to install other Java functions.
7. Use Netscape Navigator's Java switching utility to switch to Java 1.1. Open the **Netscape** folder and double click on **Java Version Selection for Netscape Navigator**. Select the appropriate Java version and click on **OK**.
8. Install TCP/IP Version 4.1 for OS/2. A new version of MPTS (Version 5.3) and new TCP/IP applications will be installed.
 - When asked whether to install a more recent version of MPTS, select **Yes**. Your current MPTS configuration will be preserved, but MPTS drivers and protocols will be updated.
 - After installation of the most recent version of MPTS, click on **Exit**.
 - When asked whether to update your CONFIG.SYS file, click on **OK**.
 - Do not shut down or restart your computer after exiting MPTS. Continue with the rest of TCP/IP Version 4.1 installation.

- After installation of MPTS, Netscape Navigator starts. Follow the instructions provided to complete installation. Shut down and reboot your PC when TCP/IP Version 4.1 installation has finished.
9. Apply the latest TCP/IP 4.1, MPTS 5.3, and Java fixes.

Note: You do not need to provide Java fixes if Java 1.1.6 was installed.

The installation of TCP/IP Version 4.1 for OS/2 creates a TCP/IP Shadows folder on your desktop.

2.4.1.2 OS/2 Warp 4 DHCP Client Configuration

DHCP client configuration for OS/2 Warp 4 depends on the version of TCP/IP you are running.

- TCP/IP Version 4.1 for OS/2
 1. Open the **TCP/IP Shadows** folder, then the **TCP/IP Configurations** folder. Double-click on the **TCP/IP Configuration (Local)** icon, or just type `TCPCFG2` at an OS/2 command prompt. The TCP/IP Configuration notebook appears.
 2. Click on the **Network** tab. Highlight the appropriate LAN interface from the list (normally **LAN interface 0**) and make sure the **Enable Interface** checkbox is checked. Then, select **Automatically, Using DHCP**, as shown in Figure 19 on page 40. Click on **OK** to close the TCP/IP configuration notebook and save this change.

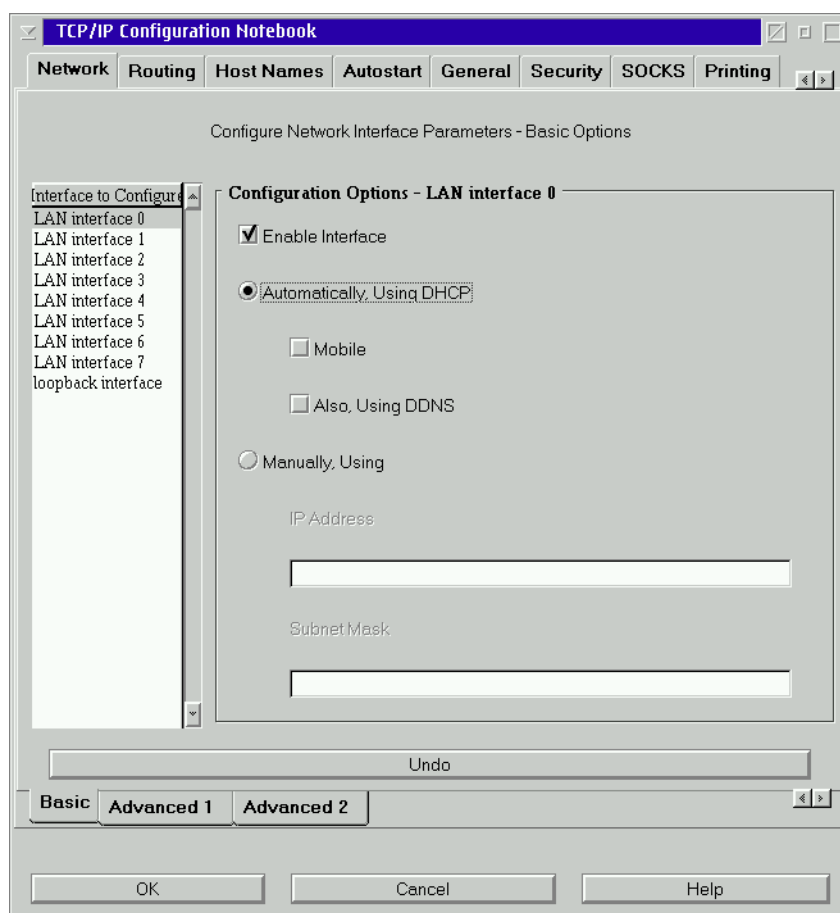


Figure 19. [OS/2 Warp] Enabling a DHCP Client with TCP/IP 4.1

3. Restart your computer to activate the DHCP client function. You can view the current IP configuration by using the DHCP client monitor utility. Open the **OS/2 System** folder, then the **System Setup** folder. Double-click on the **DHCP Monitor** icon. (Or, type `DHCPMON` at an OS/2 command prompt.) Select [**View — Details**] for more information on the lease assignment. (See Figure 20 on page 41.)

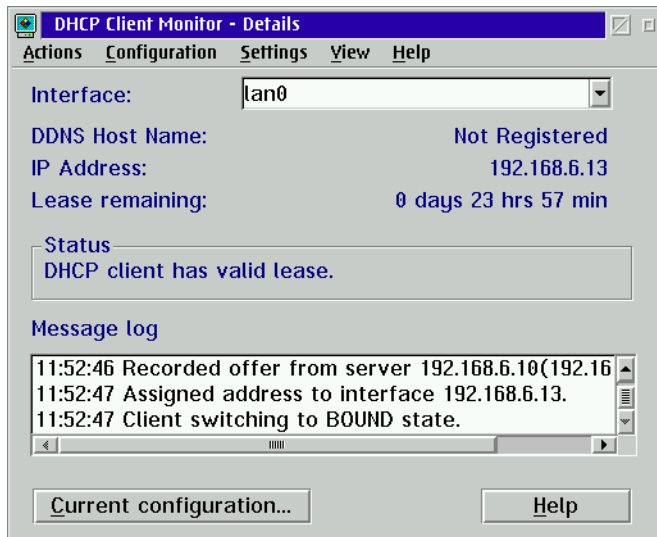


Figure 20. [OS/2 Warp] DHCP Client Monitor Using TCP/IP 4.1

You can also view all the parameters received from the DHCP server by selecting **Current Configuration** from the **Configure** menu.

DHCP Client Monitor can obtain or release an IP address assignment from any DHCP server. Select [**Action — Release lease**] to release an IP address and select [**Action — Request lease**] to obtain an IP address. (You may wish to test a new DHCP server using the DHCP Client Monitor.)

- TCP/IP 4.0

1. Open the **Programs** folder then the **TCP/IP Internet (LAN)** folder. Double-click on **TCP/IP Configuration (LAN)**. (Or, just type `TCPCFG` at an OS/2 command prompt.) The TCP/IP Configuration notebook appears.
2. Click on the **Network** tab. Highlight the appropriate LAN interface from the list (normally **LAN interface 0**) and make sure the **Enable interface** checkbox is checked. Then, select **Automatically, Using DHCP**, as shown in Figure 21 on page 42.

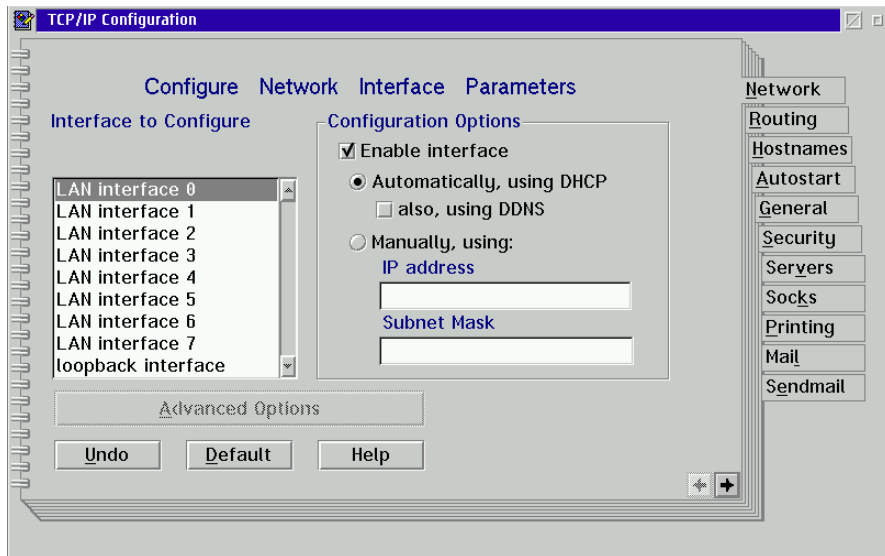


Figure 21. [OS/2 Warp] Enabling a DHCP Client Using TCP/IP 4.0

3. Close the TCP/IP configuration notebook to save this change.
4. Shutdown and restart your PC to initiate the DHCP client. You can view the current IP configuration by using the DHCP Client Monitor. Open the **OS/2 System** folder, then the **System Setup** folder. Double-click on the **DHCP Monitor** object. (Or, just type `DHCPMON` at an OS/2 command prompt.) Select [**View — Details**] for more information. (See Figure 22 on page 43.)

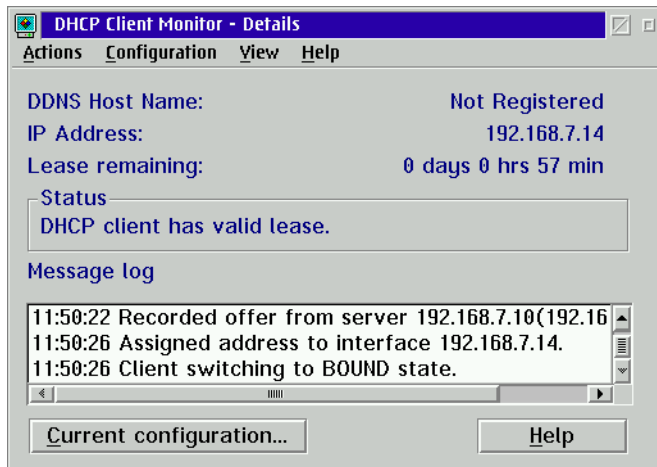


Figure 22. [OS/2 Warp] DHCP Client Monitor Window

You can also view all the parameters received from the DHCP server by selecting **Current Configuration** from the **Configure** menu (See Figure 23).

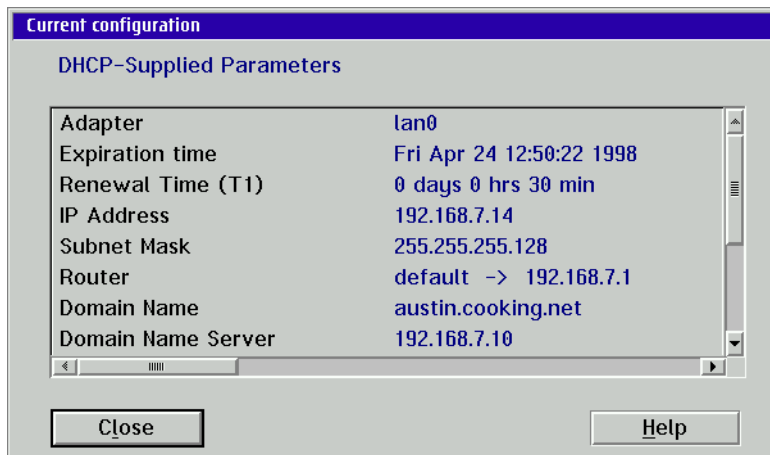


Figure 23. [OS/2 Warp] Current Configuration Window

2.4.2 Windows 95

Before running your Windows 95 computer as a DHCP client, make sure that you have installed the following components:

- TCP/IP protocol
 - An appropriate network adapter device driver
1. To set up your Windows 95 computer as a DHCP client, you need to configure the TCP/IP protocol. Double-click the **Network** icon in the **Control Panel**.
 2. In the Network notebook, highlight **TCP/IP** protocol and click on the **Properties** button. The TCP/IP properties notebook opens up. Click on the **IP Address** tab and check **Obtain an IP address automatically**, as shown in Figure 24.

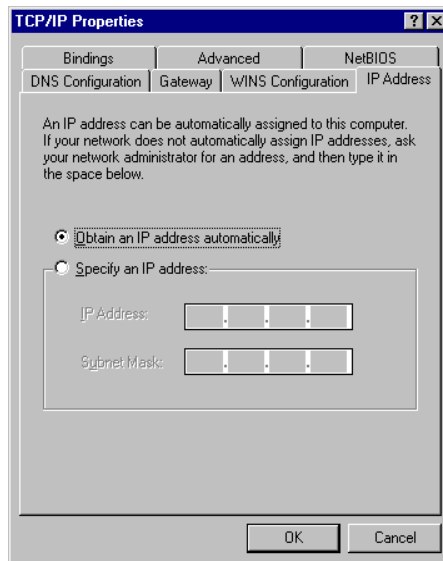


Figure 24. [Windows 95] Enabling DHCP Client

3. Click **OK** to finish the Network settings and reboot your system. Now, the Windows 95 computer should be able to obtain an IP address from the DHCP server.
4. You can view the current IP configuration by using the `WINIPCFG` command at the command prompt. This will show you a current IP address and other TCP/IP configurations, as shown in Figure 25 on page 45. This also enables you to release and renew the IP address by clicking the buttons.

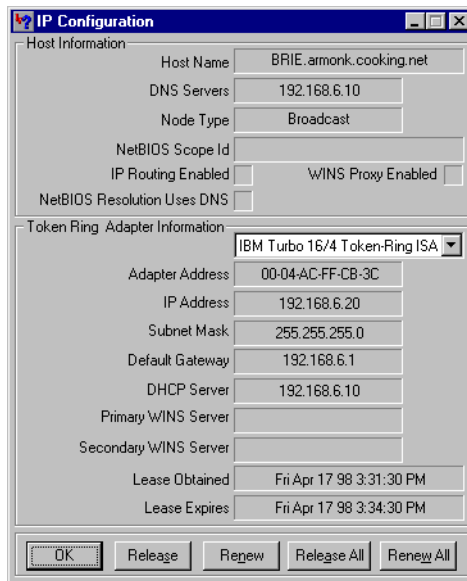


Figure 25. [Windows 95] IP Configuration Window

2.4.3 Windows NT Workstation 4.0

Before running your Windows NT Workstation computer as a DHCP client, make sure that you have installed the following components:

- TCP/IP protocol
- An appropriate network adapter device driver

To set up the Windows NT Workstation computer as a DHCP client, follow the steps below:

1. Open the **Network** icon. To open the Network icon, select [**Start** — **Settings** — **Control Panel**]. Double-click on the **Network** icon.
2. Click on the **Protocols** tab, highlight **TCP/IP Protocol** in the Network Protocol list, and click on the **Properties** button. The TCP/IP properties notebook comes out. To set your computer as a DHCP Client, click on the **IP Address** tab and check **Obtain and IP address from a DHCP server**, as shown in Figure 26 on page 46.

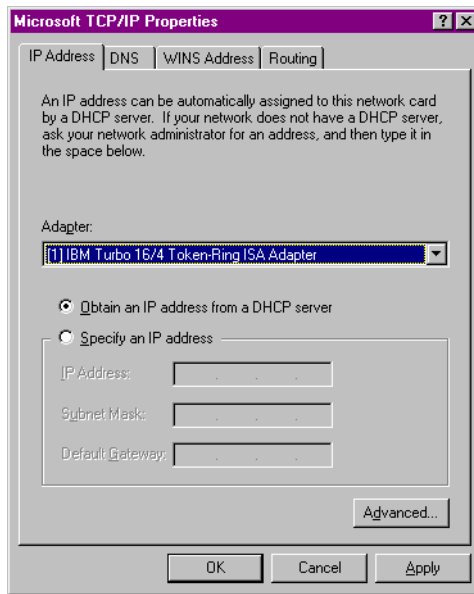


Figure 26. [Windows NT] Enabling a DHCP Client

3. Click **OK** to finish the Network settings and reboot your system. Now your Windows NT Workstation should be able to obtain an IP address from the DHCP server.

You can view the current IP configuration by using `IPCONFIG` utility. Type

```
IPCONFIG /ALL
```

at the command prompt. This will show you the current IP configuration for the operating system and the network adapter, as shown in Figure 27 on page 47.

```

C:\>ipconfig /all

Windows NT IP Configuration
    Host Name. . . . . : burger.bellevue.cooking.net
    DNS Servers. . . . . : 192.168.8.10
    Node Type. . . . . : Broadcast
    NetBIOS Scope ID . . . . . :
    IP Routing Enabled . . . . . : No
    WINS Proxy Enabled . . . . . : No
    NetBIOS Resolution Uses DNS: No

Token-Ring adapter IBMTOK1:
    Description. . . . . : IBM Turbo 16/4 Token-Ring ISA Adapter
    Physical Address . . . . . : 00-04-AC-FF-CB-3C
    DHCP Enabled . . . . . : Yes
    IP Address . . . . . : 192.168.8.12
    Subnet Mask. . . . . : 255.255.255.0
    Default Gateway. . . . . : 192.168.8.1
    DHCP Server. . . . . : 192.168.8.10
    Lease Obtained . . . . . : Thursday, April 30, 1998 10:36:31 AM
    Lease Expires. . . . . : Friday, May 01, 1998 10:36:31 AM

C:\>

```

Figure 27. [Windows NT] Displaying Current Configuration

To give up the current IP lease, type

```
IPCONFIG /RELEASE
```

at the command prompt.

To obtain a new IP address, or to update options and lease time, type

```
IPCONFIG /RENEW
```

at the command prompt.

2.4.4 Apple Macintosh

The Apple Macintosh was one of the first microcomputers to feature built-in networking, and its designers have focused on simplicity to help make the system easy to set up and use on a network. The Macintosh can easily receive its IP address assignment from any DHCP server.

We tested an Apple Macintosh running MacOS Version 8 with its built-in Open Transport TCP/IP software. (Open Transport Version 1.1.1 or later should be installed on previous versions of MacOS to support DHCP.) Either the Internet Setup Assistant or the TCP/IP Control Panel (Figure 28 on page 48) can be used to select DHCP.

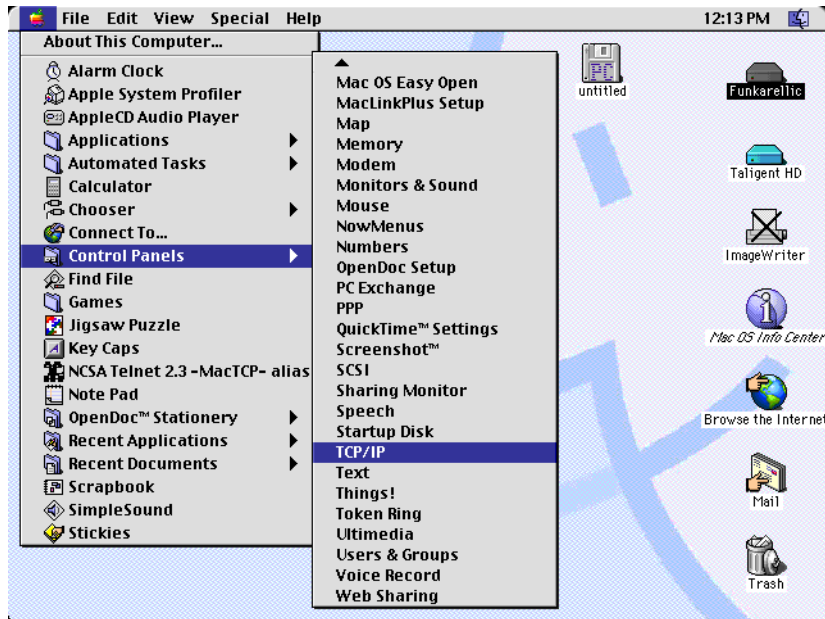


Figure 28. [Macintosh] Selecting the TCP/IP Control Panel

If you select the TCP/IP control panel a window appears, as shown in Figure 29.

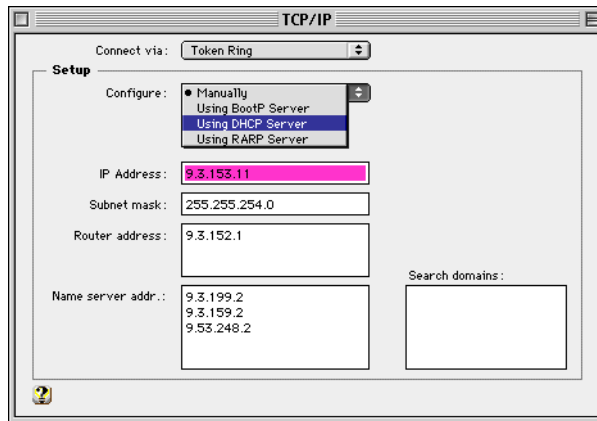


Figure 29. [Macintosh] Choosing the DHCP Server Option

Simply select **Using DHCP Server** and close the window to activate the Macintosh's DHCP client support. (Click on **Save**, as shown in Figure 30 on

page 49, when prompted.) After a system restart, you can revisit this window to examine the address assignment received from the DHCP server.

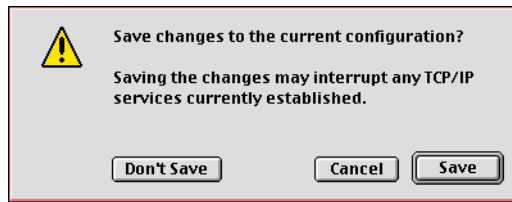


Figure 30. [Macintosh] Saving TCP/IP Configuration Changes

To further integrate your Macintoshes into your network, you may wish to consider two additional products:

- **IBM LAN Server for Macintosh** allows Warp Server to act as an AppleShare file and print server. The Macintosh's built-in Chooser can be used to access shared drives and printers on Warp Server without adding any additional software to the clients. In addition, Warp Server's Postscript translation services (Advanced Printing) can accept Macintosh Postscript print jobs and format them for PC printers in PCL, IBM PPDS, or AFP formats, providing Macintosh users access to the same printers as your PC users. For more information on LAN Server for Macintosh, please consult *Network Clients for OS/2 Warp Server*, IBM Publication No. SG24-2009. A link to this IBM redbook is available at <http://www.software.ibm.com/os/warp/library> on the Internet.
- **DAVE**, produced by Thursby Software Systems, is client software for Macintoshes designed to communicate with Warp Server, Windows NT, Windows 95, IBM DOS LAN Services, and other NetBIOS-based networks. However, DAVE must be installed on each client. For more information on DAVE, please visit <http://www.thursby.com> on the Internet.

2.4.5 IBM Network Station

We used an IBM *Network Station* computer (NC) model 200 (which is very similar to a model 100 with regards to operation and configuration of the device). We tested the BOOTP client function and the DHCP client function with the TCP/IP Version 4.1 DHCP Server.

2.4.5.1 BOOTP

First, we tested the BOOTP function. The NC was configured so that DHCP is disabled, with BOOTP as the first choice for booting. The configuration was saved, and the NC was powered off.

For this test, we want the TFTP daemon and the portmapper running on the server. The NC needs to download its kernel file and uses TFTP for that task. We configured these using `TCPCFG2` (or double-clicking on the **TCP/IP Configuration** object). As shown in Figure 31, choosing the `Autostart` page, we chose `inetd`, `tftpd`, and `portmapper`.

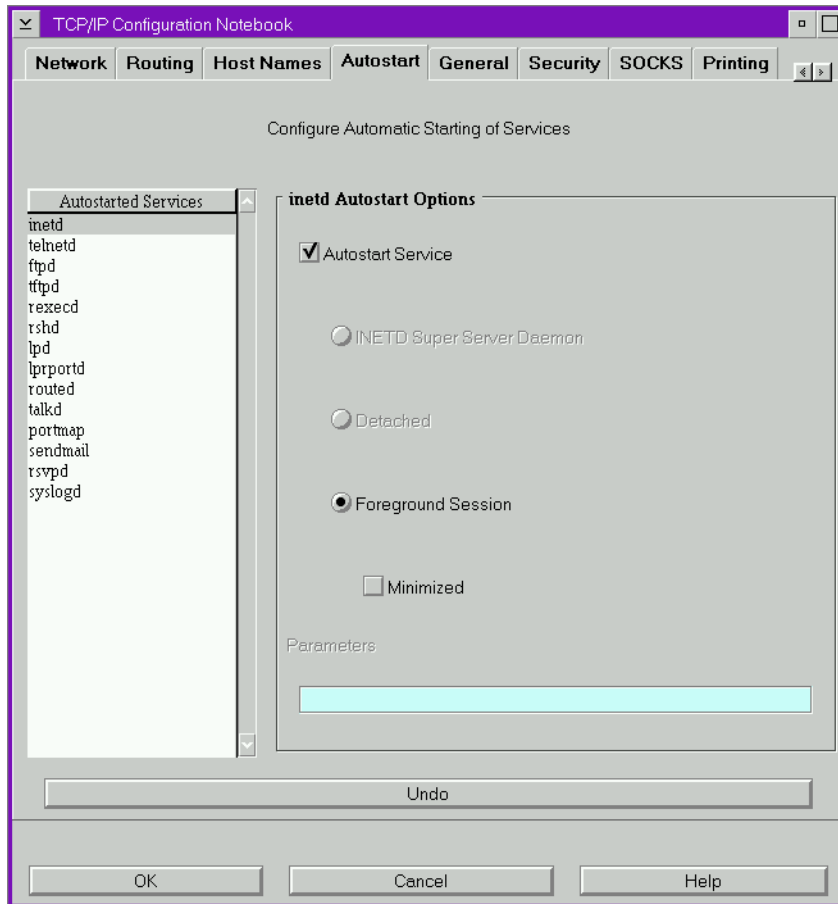


Figure 31. [Warp Server] TCP/IP Configuration Notebook: Autostart Service

We also decided to start the `syslog` daemon, as shown in Figure 32 on page 51.

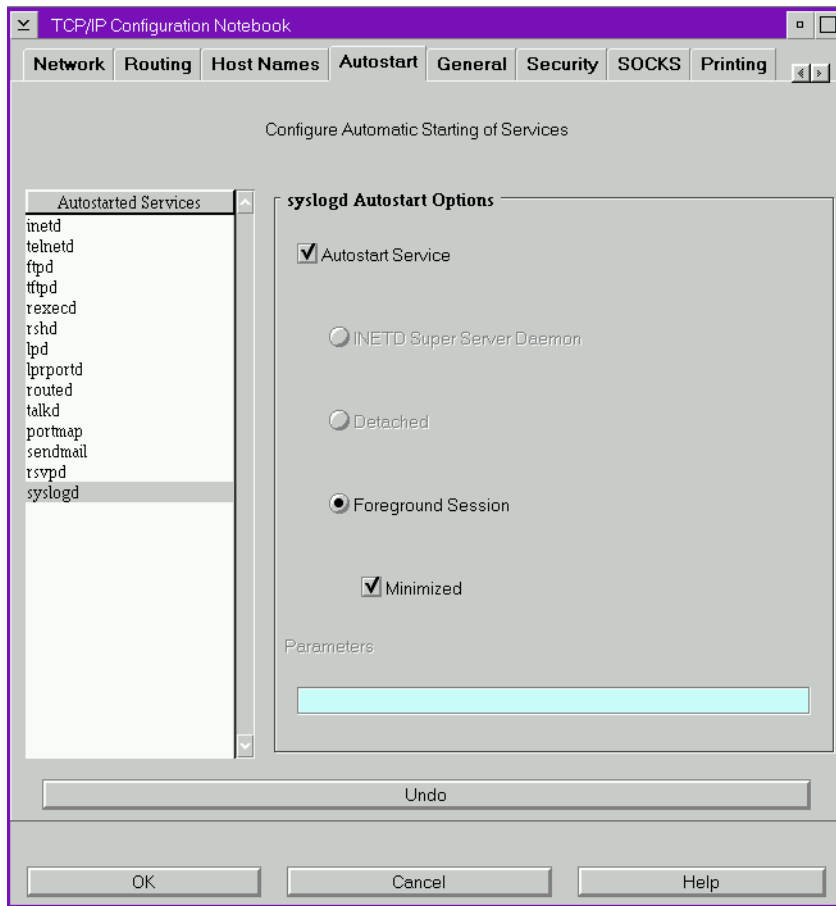


Figure 32. [Warp Server] TCP/IP Configuration Notebook: Syslog Daemon

It is useful to log activity on your servers, especially when you are using TFTP. You may wish to click the **Minimize** box, rather than leave some screens to start open as we did.

Configuring the DHCP server on OS/2 was fairly simple. We started the DHCP server configuration and highlighted the subnet we created earlier. Click on **Configure** and then **Add Client**. A notebook opens up, as shown in Figure 33 on page 52.

The screenshot shows the 'Client Definition' tab of the [Warp Server] Client Definition Settings Notebook. The form is divided into two main sections: 'Name' and 'Lease time and comment'.

Name Section:

- Client name: NetWorkSta
- Client hardware type: 6 IEEE 802 Networks
- Client ID: 0000e5687160
- IP Address: 192.168.6.17
- Buttons: Assign any address, Assign this address

Lease time and comment Section:

- Default lease time: 24 hours
- Enter a lease time: 0 years, 0 months, 0 days, 0 hours, 0 minutes, 0 seconds
- Permanent lease: (radio button)
- Comment: IBM Network Station 200 (aka 100)

At the bottom right, there are buttons for OK, Cancel, and Help.

Figure 33. [Warp Server] Client Definition Settings Notebook

We filled it in with our chosen values on the Client Definition page. Note that the client ID is the MAC address of the device in question. On the DHCP Options page, we chose **option 67, Boot File Name**, so that we could boot the NC. Note that no path is listed for the boot file, as shown in Figure 34 on page 53.

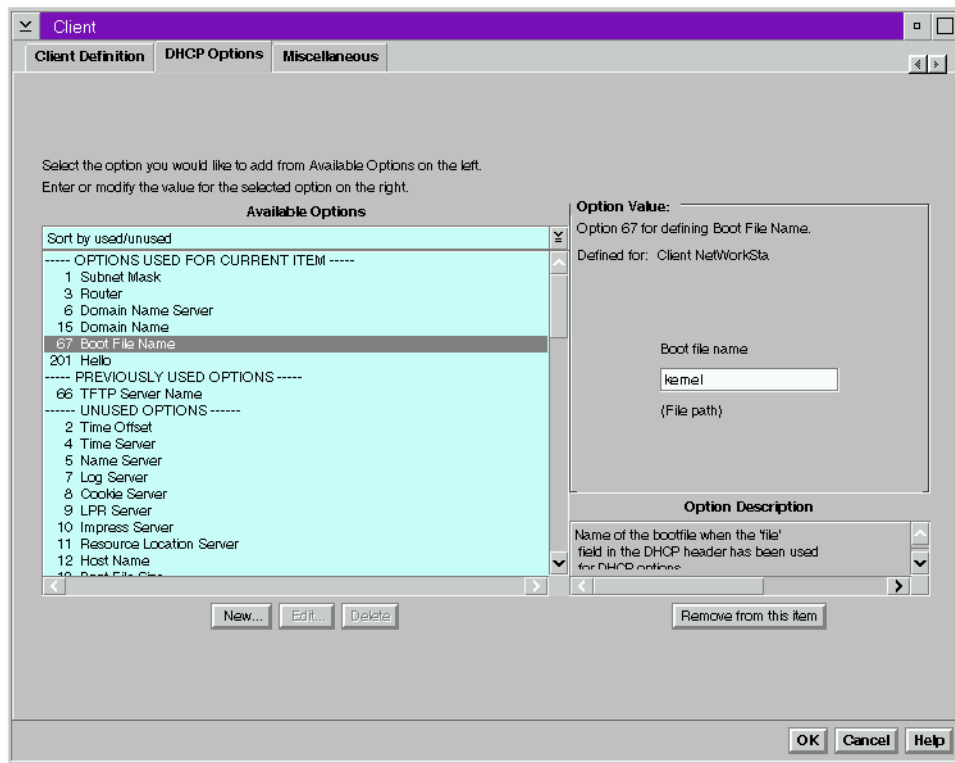


Figure 34. [Warp Server] DHCP Options Settings Notebook

We configured TFTP security for this client, and the home directory is C:\TFTPBOOT. The directory tree for the NC is located in the C:\TFTPBOOT directory. The following line was placed in the C:\MPTN\ETC\TFTPPAUTH file:

```
C:\TFTPBOOT RO 192.168.6.17
```

Note that the IP address matches the IP address Reserved for the NC when we performed the Add Client, as shown in Figure 33 on page 52.

Options 1, Subnet Mask, and 15, Domain Name, are inherited from our subnet settings. Now, we turn on BOOTP client support by double-clicking **DHCP Server**, choosing the **Miscellaneous** page, and checking **Support BOOTP clients**, as shown in Figure 35 on page 54.

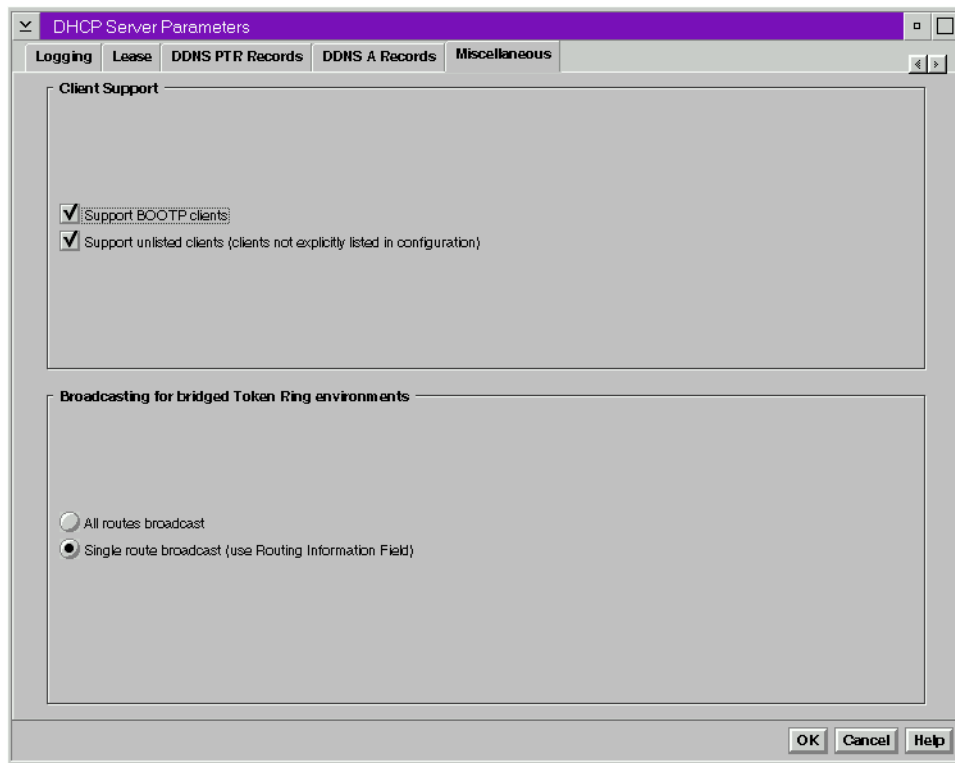


Figure 35. [Warp Server] Support BOOTP Settings Notebook

After saving the notebook, the DHCP server was reinitialized by starting **DHCP Server Management** and choosing **Re-initialize**. From an OS/2 command line, we executed the following command:

```
DADMIN -S | FIND /I "-"
```

to see the in-use addresses, then we turned on the NC. When going back and adding options like this, remember to save your file after you have saved a notebook and to re-initialize the DHCP server after changing or adding options. You might, as we did, start to wonder why your new options are not working!

Windows NT can provide the BOOTP client an address and a file name, but it cannot provide the boot file. Neither NFS nor TFTP daemon is delivered with Windows NT; so our IBM Network Station was not able to receive its boot file from Windows NT. However, Windows NT can provide the IP address of a boot file server. For fun, we set up the Windows NT DHCP server to provide the IP address of one of our OS/2 DHCP clients as the boot file server. We

configured the OS/2 client to autostart the inet daemon, TFTP daemon (started from inetd), and the portmapper. We rebooted the OS/2 DHCP client to restart everything. We configured the NT DHCP server so that the boot file server provided to the BOOTP client (NC) was the address of the OS/2 client. After restarting the NT DHCP server, we powered on the NC and it booted up from the OS/2 workstation. So, the Windows NT DHCP server provides BOOTP services, kind of. If you want more than just an address (for example, you actually want to network boot a workstation), you will need to provide the name of a boot file server, or add some third party software to Windows NT; so it can provide TFTP services (we did not test any third party add-ons). Note that IBM has Networkstation kits for different server platforms that provide file transfer services.

2.4.5.2 DHCP

Next, we configured the NC as a DHCP client. No changes on either server were required to boot the NC as a DHCP client

We found one caveat on the NC which may be due to older firmware in our machine. On at least one occasion, even though the NC was configured to get its IP address from the network, it insisted on using the IP address that was configured in NVRAM. We solved this problem by setting the NVRAM addresses to all zeroes, except for the router and subnet addresses.

2.4.6 Linux

UNIX clients can certainly participate in a DHCP network, getting address assignments from a DHCP server, such as OS/2 Warp Server.

Using instructions available on the Internet at:

<http://www.cro.net/~vuksan/dhcp.html>

we tested RedHat Linux 5.0 with kernel 2.0.31. We dubbed this system woody. Woody succeeded in picking up a dynamically assigned IP address using either BOOTP or DHCP (both worked). However, getting DHCP to work properly with woody's token-ring adapter required a patch to the dhcpd program file. This patch is available at:

<http://eolicom.olicom.dk/~storer/dhcp>

on the Internet. Although a couple warning messages were displayed while woody tried to pick up its DHCP assignment, apparently because some unsupported options were conveyed, the system managed to get connected to the TCP/IP network without any added difficulty.

2.4.6.1 Step-by-Step Procedure

To enable RedHat Linux's DHCP, follow these steps:

1. Log on as user root.
2. If needed, replace the dhcpd program file in the /sbin directory with the corrected version required for token-ring networks. Make sure that the user and group IDs for dhcpd are both set to root, and the permissions are set to read (r) and execute (x) for all users.
3. If not running already, start X-Windows with the command:

```
startx
```

4. Start the control panel, as shown in Figure 36, with the following command:

```
control-panel
```



Figure 36. [RedHat Linux] Control Panel

5. Click on the network configuration section (bottom or right most icon).

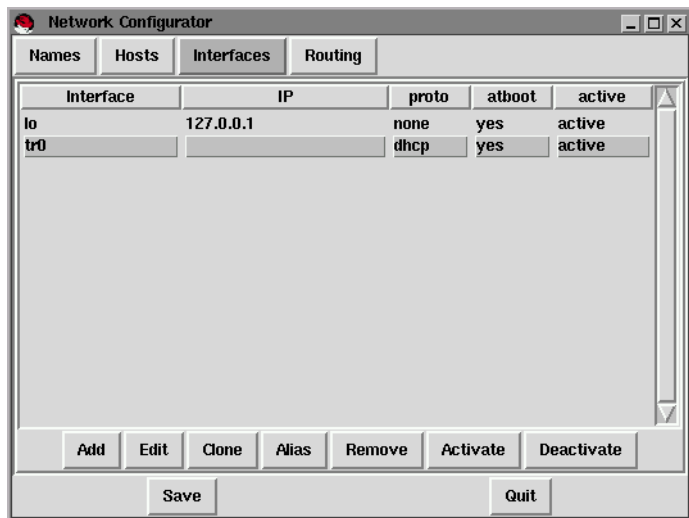


Figure 37. [RedHat Linux] Interfaces Section of Network Configuration

6. In the Interfaces section (see Figure 37 on page 56), edit tr0 (for token-ring) or eth0 (for Ethernet). Change the interface configuration protocol (see Figure 38) to DHCP (or BOOTP), then click on **Done** to save these changes.

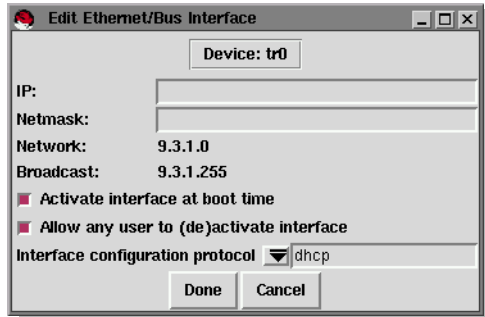


Figure 38. [RedHat Linux] Editing the tr0 (Token-Ring) Interface

7. Click on **Save**, then **Quit**, to exit out of the network configuration section.
8. Shutdown and restart the Linux system using the command:

```
shutdown -h now
```

9. After rebooting, and after logging in, verify your address assignment by using the command:

```
ifconfig
```

You can also verify network connections by using the ping command.

Linux is supported by its community of users and developers in many popular Internet newsgroups, including comp.os.linux.answers.

2.4.7 DOS, Windows 3.1, and Windows for Workgroups

In our test environment, we set up three DOS-based environments: DOS, DOS with Windows 3.11, and DOS with Windows for Workgroups. The DOS used was IBM's PC-DOS 2000, the only fully supported Year 2000 Ready version of DOS.

All three environments, then, had IBM's DOS LAN Services (*DLS*) installed. DLS is the DOS LAN client supplied with Warp Server, and it comes with a TCP/IP stack from Network TeleSystems that supports DHCP.

Before installing DLS, make sure that you have all four diskettes and that you are suitably licensed.

2.4.7.1 Step-by-Step Procedure

To install DLS and enable TCP/IP, follow these steps:

1. Insert diskette 1, and enter the following command:
A: INSTALL
2. At the opening screen, press **ENTER** to continue
3. When prompted to choose directory, accept the default of C:\NET
4. When prompted to enter the machine ID, enter the machine's name. In this instance, we are entering the NetBIOS name.
5. When prompted to enter the user name, enter the user ID of the person who would normally use this machine.
6. When prompted to enter the domain name, enter the name of the domain that the user would normally log onto.
7. The options panel will be displayed. See Figure 39.

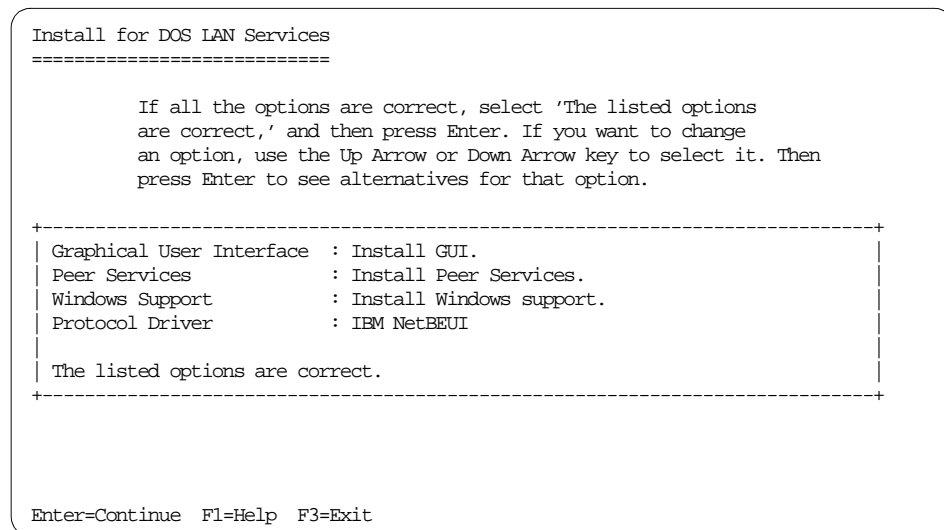


Figure 39. [DOS LAN Services] Installation Options

8. If you do not require Peer support, scroll up to the **Peer Services option**, and press **Enter**, then select the option **Do not install Peer services**.
9. If you do not require Windows support, select the **Windows Support option**, then select the option **Do not install Windows support**.
10. Select the **Protocol Driver option**.
11. Select the **Change driver for protocol option**.

12. A panel will be displayed showing the protocols available for DLS, as shown in Figure 40.

If you are using native DOS, select **TCPBEUI (Real-Mode) & IBM NetBEUI**

If you are using Windows 3.11 or Windows for Workgroups, select **TCPBEUI (Windows Protect-Mode) & IBM NetBEUI**

We recommend dual protocols, as NetBEUI is still the most efficient within a LAN. TCPBEUI is required to activate TCP/IP and can be used for WAN (wide area network) communications.

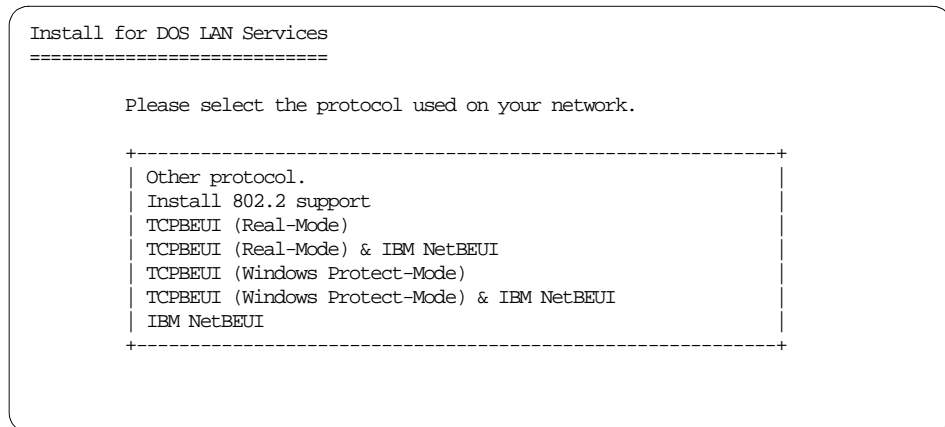


Figure 40. [DOS LAN Services] Selecting a Protocol

13. When prompted for the IP address with the `IPAddr=` prompt, do not enter an address, just press **Enter**.

A red error panel will be displayed (See Figure 41 on page 60), stating that you have not specified a required value. Ignore the error, and press **C** to continue without setting the value.

Install for DOS LAN Services

=====

The IP Address (IPAddr) is a required parameter for the protocol you have chosen. This address is unique to your computer. An example of an IP address would be:

IPAddr=123.2.43.7

```
+-----+
|                                         |t.)
| You did not specify the value of a required setting. |
|                                         |
| To specify its value, press Enter.      |
| To continue Install without setting its value, press C. |
|                                         |
+-----+
```

IPAddr=_____

Figure 41. [DOS LAN Services] Error When Required Value Not Specified

14. When prompted for the NetSubNetMask, GatewayAddr, and DNSAddr parameters, do not enter values just press **Enter**. As before, press **C** at the error panel to continue without entering a value.
15. Select **Edit settings for protocol driver**. A panel with the configurable options will be displayed, as shown in Figure 42 on page 61.

Important

It is not readily apparent that the configurable options panel is only showing a portion of the options. More options are visible if you scroll up the list.

Install for DOS LAN Services

=====

The settings for your protocol driver are listed below. If all the settings are correct, select 'The listed options are correct'. Then press Enter. If you want to change a setting, use the Up Arrow or Down Arrow key to select it. Then press Enter to see alternatives for that setting.

Protocol Driver : TCPBEUI (Real-Mode) & IBM NetBEUI

```
+-----+
| ARP Cache Timeout in seconds=300
| DHCP Client ID=
| Virtual Circuits=16
| VC Receive Buffers=6
| VC Send Buffers=6
| Receive Window in Bytes=2920
| Send Window in Buffers=8
| Use Memory Option=UMB
|
| The listed options are correct
+-----+
```

F1=Help F3=Exit Esc=Previous Screen

Figure 42. [DOS LAN Services] Options for the Protocol Driver

16. Scroll up the list to the `Bootp=` option, and press **Enter**, then select **DHCP** from the list of options.
17. Select the **DHCPClientID=** option and enter the machine ID, as used during step 4 on page 58. In this instance, we are entering the computer's TCP/IP hostname.
18. Press **Enter** on **The listed options are correct, Driver configuration is correct**, and **The listed options are correct** options, as they are displayed.
19. Select the **Redirector**.

For native DOS clients, select either **Use the basic**, **Use the full**, or **Use the protect mode redirector** options.

For Windows 3.11 or Windows for Workgroups clients, select **Use the virtual redirector**.
20. Select the **Startup** option and then select **Run DOS LAN Services only**.
21. If necessary, select **Network Card** and choose the network adapter that you use. See Figure 43 on page 62.

Install for DOS LAN Services

=====

Select the type of network card that is installed in your computer, and then press Enter.

```
+-----+
| IBM 16/4 Token Ring Credit Card Adapter
| IBM Token Ring (All Types)
| IBM Token Ring (MCA)
| IBM Token Ring II
| IBM Token Ring 4/16Mbps
| IBM Token Ring 4/16Mbps (MCA)
| IBM Token-Ring Network 16/4 Adapter II
| Intel EtherExpress 16 or 16TP
| Intel TokenExpress EISA 16/4
| Intel TokenExpress 16/4
| Novell/Anthem NE2000
| Novell/Anthem NE/2
| Olicom 16/4 Token-Ring Adapter
| Proteon Token Ring (P1392)
| SMC (WD) EtherCard PLUS/A (MCA) (WD 8003E/A or 8003ET/A)
| SMC (WD) EtherCard PLUS Elite 16 Combo (WD/8013EW or 8013EW)
+-----+
```

Figure 43. [DOS LAN Services] Network Adapter Types

22. Select **The listed options are correct.**

You will be re-prompted to enter the `IPAddr`, `NetSubNetMask`, `GatewayAddr`, and `DNSAddr` options. As before, leave these entries blank and press **C** at the error panel to continue without entering a value.

23. Select **The listed options are correct.**

24. DLS will now be installed. Change diskettes as required.

25. At the completion of the installation, you will be prompted to either press Enter to restart, or F3 to exit to DOS. Press **F3**, then use a text editor to modify the `\NET\NETWORK.INI` file.

```

[network]
computername=BURGER
lanroot=C:\NET
autologon=no
autostart=netbeui full messenger
username=USERID
domain=ARMONK
lslogon=yes
reconnect=no
passwordcaching=no
timesync=yes

```

Figure 44. [DOS LAN Services] Extract from NETWORK.INI

Figure 44 shows an extract from the NETWORK.INI file, with modifications that we have made to the defaults highlighted. Note: The autostart parameters will vary depending on the Redirector option chosen in step 19.

26. Save the file, then restart the computer. As it reboots, you should see messages similar to those in Figure 45.

```

IBM Protocol Manager Version 2.1
IBM Token-Ring Network Driver Version2.7

NISTS: Initialization started
Installing Network TeleSystems TCP/IOP
Version: 2.09 (960320)
Using VCs=16 VCSends=6 VCReceiveLarge=6
Driver NTSTS$ in section [NTS$NIST2] will bind with module [IBM$GENIBMTOK]
NTS$NISTS2: Initialization succeeded
IBM DOS LAN Services Driver Version 5
Copyright (C) International Business Machines Corporation 1993, 1995
Copyright (C) Microsoft Corporation 1992
    Transport Hooks Enabled
SHARE installed
IBM Netbind Version 2.1
Waiting for DHCP Server

IBM DOS NETBEUI 3.00
Copyright (C) International Business Machines Corporation 1993, 1995
Copyright (C) Microsoft Corporation 1992
The command completed successfully

C:\>

```

Figure 45. [DOS LAN Services] Messages Displayed as a DOS Workstation Starts

27. You can now test your IP configuration, by using the `PING` command (either in native DOS, or by opening a DOS window from Windows). For example, if we had used the TCP/IP name `BURGER` in step 17, we would issue the command `PING BURGER`.

```
C:\>PING BURGER
PING - ICMP Echo Request/Reply 2.09 (960320).
Copyright (c) 1995 Network TeleSystems, Inc. All rights reserved.
PING burger (192.168.6.16): 56 ICMP data bytes
64 bytes from burger: icmp_seq = 0. time < 55 ms

----burger PING Statistics ----
1 packets transmitted
1 packets received
0% packet loss
round-trip (ms) min/avg/max = 55/55/55

C:\>
```

Figure 46. [DOS LAN Services] Testing IP Configuration Using `PING`

2.4.8 Sun Microsystems Solaris

The Sun Microsystems UltraSparc workstation was simple to set up as a DHCP client. We wanted to use a token-ring card in our UltraSparcstation but we could not. Remember in our Linux setup, we told you about a DHCP bug we found when using a token-ring interface. We found a similar bug in the Sun code. Sun is aware of this and is working on it. If you really need a token-ring adapter in your UltraSparc workstation, you should contact Sun. Our UltraSparc workstation came pre-installed with SunOS version 5.6.

Before getting started on the client, we configured our DHCP server to serve addresses for the 192.168.9.0 subnet. In the beginning of this chapter, we showed you how to add a subnet to your DHCP server.

Since the token-ring would not work with DHCP on the Sun machine, we had to revert back to the Ethernet interface. To do this we executed the following command:

```
/usr/sbin/ifconfig hme0 plumb
```

We then configured a router between our Ethernet network (the lone Sun machine) and our token-ring networks (everything else, we have described in this book), as shown in Figure 47 on page 65.

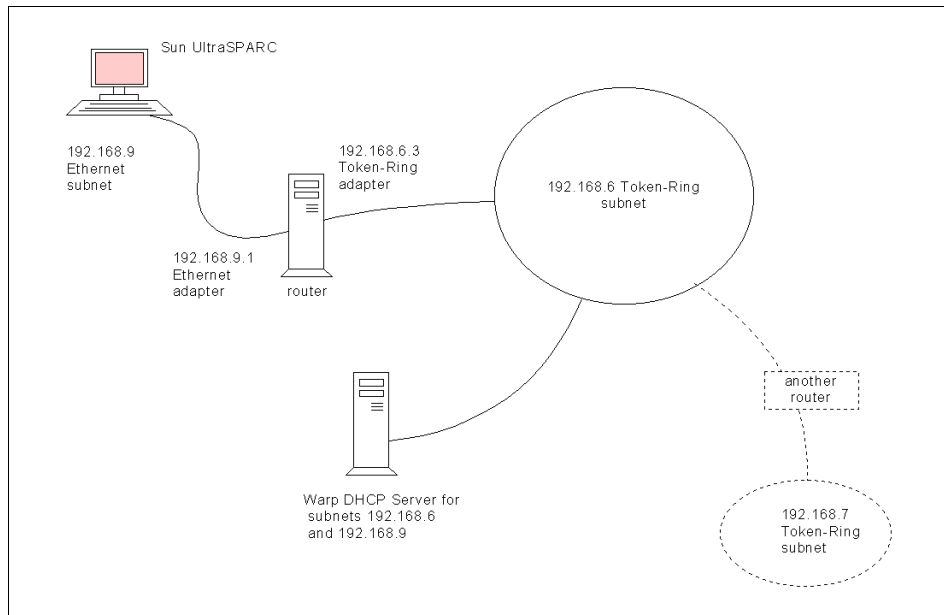


Figure 47. Integrating the Sun Solaris Workstation into Our Network

To verify that this configuration was functional, we assigned an IP address to the interface with the command:

```
/usr/sbin/ifconfig hme0 192.168.9.145
```

and configured routing:

```
/usr/sbin/route add net 192.168.9 192.168.9.145
```

```
/usr/sbin/route add default 192.168.9.1
```

We could then contact our DHCP server by executing:

```
/usr/sbin/ping 192.168.6.10
```

After we verified the router and network configuration, we then configured the UltraSparc workstation as a DHCP client. This was actually the easiest machine to configure as a DHCP client. We simply created the following files:

- /etc/dhcp.hme0
- /etc/hostname.hme0

as shown in the following command:

```
/bin/touch /etc/dhcp.hme0 /etc/hostname.hme0
```

Now, you can simply reboot the UltraSparc workstation, or you can execute:

```
ifconfig hme0 dhcp start
```

After a minute or so, our machine received the address 192.168.6.22, as shown in Figure 48.

```
/usr/sbin/ifconfig hme0
hme0: flags=4843<UP,BROADCAST,RUNNING,MULTICAST,DHCP> mtu 1500
inet 192.168.6.22 netmask fffffff0 broadcast 192.168.6.255
ether 8:0:20:7e:38:27
```

Figure 48. DHCP Results on Sun UltraSparc

The configuration gets stored in the `/etc/dhcp/hme0.dhc` file, a binary file.

You should also refer to your Sun documentation. The following `man` pages may be useful also:

- `dhcagent` (1M)
- `dhcinfo`
- `ifconfig` (1M)
- `netstat` (1M)

2.4.9 IBM WorkSpace On-Demand

Unlike other PC-based software, IBM's WorkSpace On-Demand does not require any software to be loaded on a client system. All the software a PC needs to operate is loaded from the WorkSpace On-Demand manager across the network. All management, installation of new applications, software updates, and other tasks are performed at the server. Consequently, WorkSpace On-Demand costs a lot less to manage and support than other PC-based solutions.

Because WorkSpace On-Demand supports modern Java and Web-based applications, network managers need an easy way to set up TCP/IP for each client. Managers can access WorkSpace On-Demand setup right from the server or from a remote location across the network. Turning on DHCP for a client merely involves selecting a button and clicking on Set. The WorkSpace On-Demand manager can also act as the DHCP server, if needed. Alternatively, a separate DHCP server (such as OS/2 Warp Server, AIX, or Windows NT) may provide the address information to a WorkSpace On-Demand client.

2.4.9.1 Step-by-Step Procedure

To activate DHCP for a WorkSpace On-Demand Version 1.0 client, follow these steps:

1. Log onto the WorkSpace On-Demand manager as a system administrator.
2. Open the **LAN Server Administration** program.
3. Open the **Local Workstation** icon for the WorkSpace server.
4. Open the **Remote IPL Requesters** folder.



Figure 49. [WorkSpace On-Demand] Contents of Remote IPL Requesters Folder

5. If you need to create a new icon for a new WorkSpace On-Demand client, drag the WorkSpace template to any empty part of the folder. (See Figure 49.) Otherwise, open any existing WorkSpace client icon.
6. Select the **IP Address** tab.
7. Make sure the **Configure TCP/IP** checkbox is selected.
8. Click on the **Automatic, using DHCP** radio button to turn DHCP on. (See Figure 50 on page 68.) You may also select **Also using DDNS** (see Chapter 4, "Serving Names" on page 87) if you have a Dynamic DNS server available on the network, such as Warp Server.

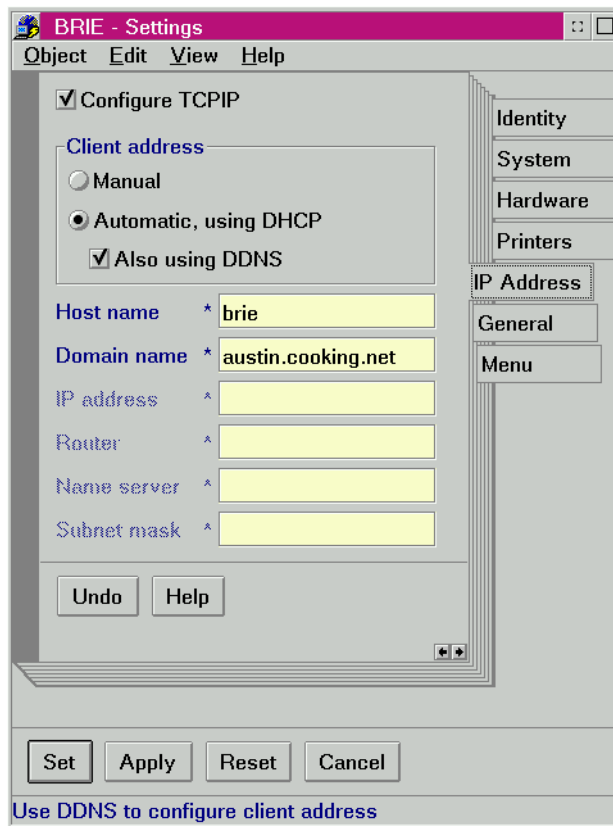


Figure 50. [WorkSpace On-Demand] IP Address Setup for Client

9. After choosing any of the other settings required elsewhere in this notebook, click on **Set** to save and close the settings for this client.

You may repeat these steps as often as needed to set up each WorkSpace client. If you have many WorkSpace clients to set up (or modify), you may automate the procedure with a REXX script.

We tested both the WorkSpace On-Demand clients and the WorkSpace manager with DHCP servers running on OS/2 Warp Server, AIX, and Windows NT. We found them to operate correctly with all these DHCP servers. (If you opt to assign a dynamic IP address to the WorkSpace On-Demand manager we recommend that you also assign a fixed hostname using a Dynamic DNS server. See Chapter 4, “Serving Names” on page 87.)

2.4.9.2 Related Publications

The IBM redbook titled *IBM WorkSpace On-Demand Handbook*, SG24-2028, provides some additional details on setting up WorkSpace On-Demand. In particular, Section B.13 describes some additional steps needed for a WorkSpace client to correctly use DDNS. Section G.5 includes information on setting up a large batch of WorkSpace clients using a REXX script.

2.4.10 Hewlett-Packard LaserJet 4000 Printer (JetDirect)

We configured the HP 4000 LaserJet Printer with JetDirect Token-Ring card. The firmware on the JetDirect Token-Ring card we used was G.05.35. This was the latest release as of the time this book was written. Please note, there are other methods of configuring addresses on JetDirect cards which are described in the Hewlett-Packard manuals shipped with the product. For example, you can use RARP. Please refer to your manuals for further information beyond what is described here.

2.4.10.1 DHCP

We set up our DHCP server to reserve an IP address for our printer. If we want to Telnet to the printer to review or change some settings, it makes it quicker and easier to do so if we already know the printer address. We reserved an address on our DHCP server for the printer named potato, as shown in Figure 51 on page 70. We used a Telnet session with our printer to configure it to act as a DHCP client. To find out the initial IP address that the printer was using, we used the buttons on the printer itself to print its configuration.

Client

Client Definition | DHCP Options | Miscellaneous

Name

Client name: potato

Client hardware type: 6 IEEE 802 Networks

Client ID: 00060D3912BE

IP Address: ☐ Assign any address ☒ Assign this address: 192.168.6.11

Lease time and comment

☐ Default lease time: 30 minutes

☐ Enter a lease time: years months days hours minutes seconds

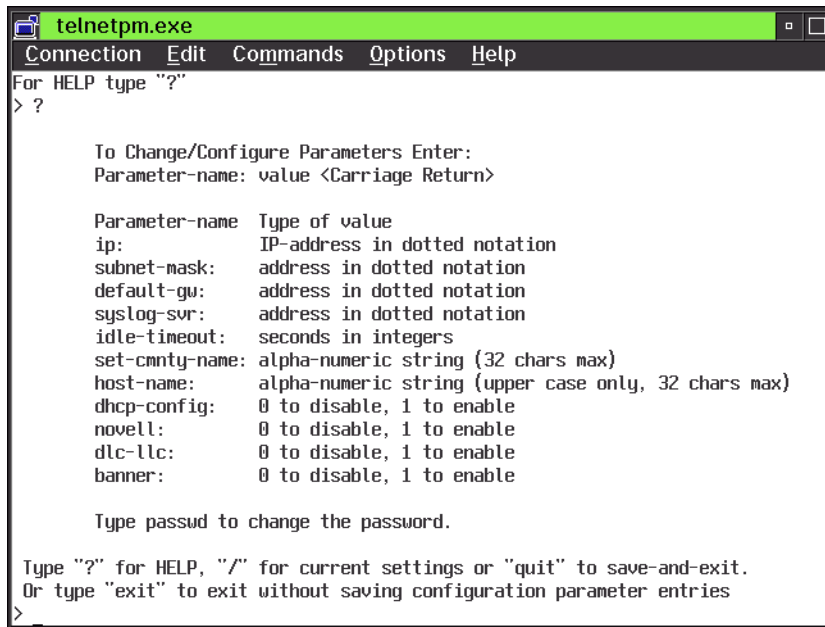
☒ Permanent lease:

Comment: Hewlett Packard LaserJet 4000

OK Cancel Help

Figure 51. [Warp Server] DHCP Printer Configuration

Figure 52 on page 71 lists commands you can execute when you Telnet into your printer.

A screenshot of a Windows command prompt window titled 'telnetpm.exe'. The window has a menu bar with 'Connection', 'Edit', 'Commands', 'Options', and 'Help'. The main text area shows a telnet session with a printer. It starts with 'For HELP type "?"' and a prompt '> ?'. Below this is a list of parameters and their types: 'ip: IP-address in dotted notation', 'subnet-mask: address in dotted notation', 'default-gw: address in dotted notation', 'syslog-svr: address in dotted notation', 'idle-timeout: seconds in integers', 'set-cmnty-name: alpha-numeric string (32 chars max)', 'host-name: alpha-numeric string (upper case only, 32 chars max)', 'dhcp-config: 0 to disable, 1 to enable', 'novell: 0 to disable, 1 to enable', 'dlc-llc: 0 to disable, 1 to enable', and 'banner: 0 to disable, 1 to enable'. Below the list is the instruction 'Type passwd to change the password.' and a final instruction: 'Type "?" for HELP, "/" for current settings or "quit" to save-and-exit. Or type "exit" to exit without saving configuration parameter entries'. The prompt '>' is at the bottom.

```
telnetpm.exe
Connection Edit Commands Options Help
For HELP type "?"
> ?

To Change/Configure Parameters Enter:
Parameter-name: value <Carriage Return>

Parameter-name  Type of value
ip:             IP-address in dotted notation
subnet-mask:    address in dotted notation
default-gw:     address in dotted notation
syslog-svr:     address in dotted notation
idle-timeout:   seconds in integers
set-cmnty-name: alpha-numeric string (32 chars max)
host-name:      alpha-numeric string (upper case only, 32 chars max)
dhcp-config:    0 to disable, 1 to enable
novell:         0 to disable, 1 to enable
dlc-llc:        0 to disable, 1 to enable
banner:         0 to disable, 1 to enable

Type passwd to change the password.

Type "?" for HELP, "/" for current settings or "quit" to save-and-exit.
Or type "exit" to exit without saving configuration parameter entries
>
```

Figure 52. Executable Commands at the HP Printer Through a Telnet Session

We executed these commands initially:

```
dhcp-config: 1
host-name: POTATO
```

We executed a Telnet session with the printer to see its configuration after it booted up from the DHCP server. This is shown in Figure 53 on page 72.

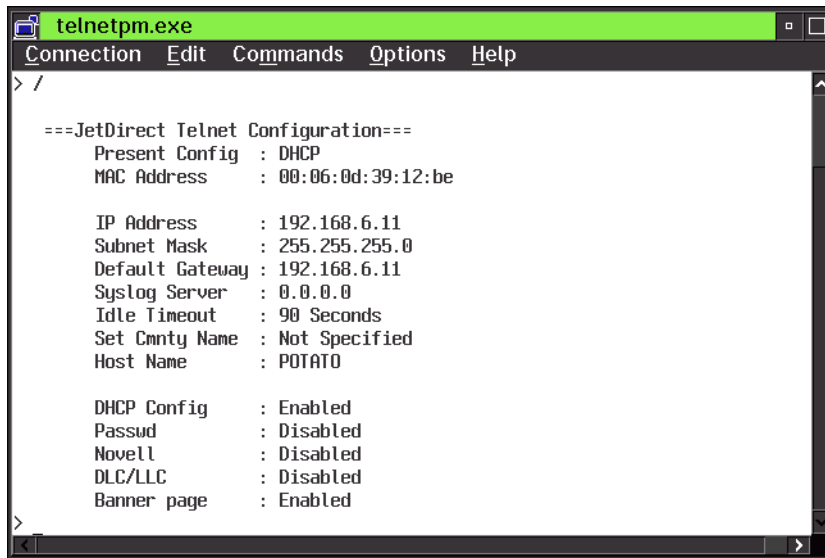


Figure 53. HP Printer Configuration after Boot Up

After configuring the DHCP server and the printer, you can use the printer directly from clients that run a line printer daemon. You can also run a line printer daemon on a Warp Server and share a printer resource with your LAN clients. In this case, the line printer daemon would run on the server itself. To share the printer with your LAN clients, follow these steps.

1. Open the **TCP/IP Configuration** GUI and select **lprportd** to start automatically. Save and exit the window.
2. Start the line printer daemon manually before creating the printer object. At an OS/2 command prompt, type the following two commands:

```
DETACH LPRPORTD
DETACH LPD
```

3. Create a printer object by dragging a printer off of the printer template.
4. Choose the correct printer driver (or install a new one) and also choose one of the **\PIPE\LPD** ports, as shown in Figure 54 on page 73.

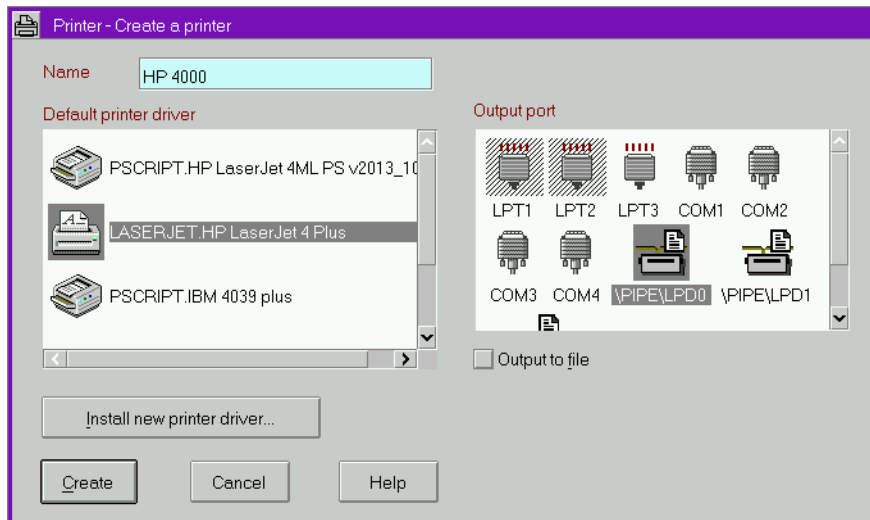


Figure 54. [OS/2 Warp] Create LPD Printer

5. Right click on your chosen **\PIPE\LPD** port after your have highlighted it. Fill in the blanks, as shown in Figure 55 on page 74, using your printer address for the LPD server. The LPD printer is either raw1 or text1 on a single port JetDirect card. If you have a multi-port JetDirect box, the hardware ports correspond to the number (for example, raw1, raw2, raw3 for a 3-port JetDirect box). Click **Create**, when you are done.

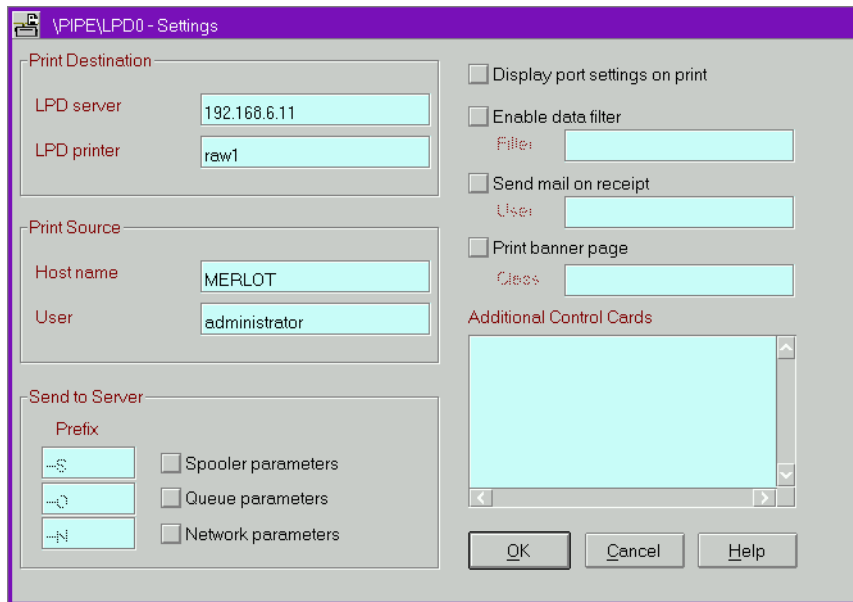


Figure 55. [OS/2 Warp] LPD Settings Notebook

Your printer can now be shared via Warp Server like any other printer resource, or you can print to it UNIX style by using `LPR`.

If you want to print directly to the printer from the server or a client, you can type the following command:

```
LPR -b -p RAW1 -s 192.168.6.11 <filename>
```

2.4.10.2 BOOTP

Officially, Hewlett-Packard does not support BOOTP on the JetDirect card, unless you are using HP JetAdmin software on the HP-UX and Sun platforms. That is what we were told when we called support for help. We were able to get the printer to work, but not completely as expected. We could not get the printer to consistently accept the IP address that we wanted it to use. We suggest that you use this device as a DHCP client, rather than a BOOTP client.

Chapter 3. DHCP Server and Client Interaction

This section describes the interaction between DHCP clients and servers, and how they communicate across a network. You may wish to skip this chapter if you are not interested in the technical aspects of DHCP communication.

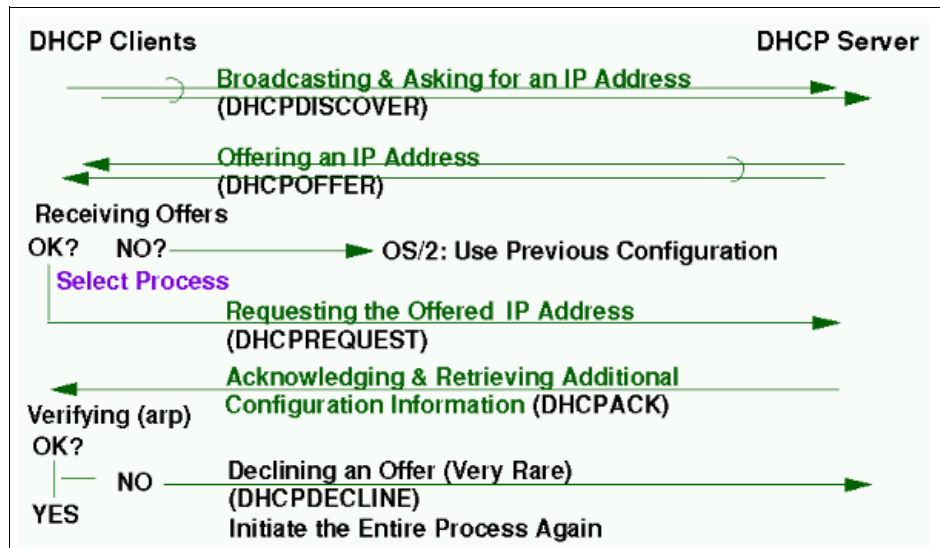


Figure 56. DHCP Client and DHCP Server Interaction

3.1 DHCP Initialization and Acquisition Process

If a client uses multiple IP interfaces, each of them must be configured by DHCP separately. The following steps show how a DHCP client is initialized. The screen shots are of Network Associates Sniffer Basic product (formerly known as NetXRay). The entire series of packets are fully decoded in a text file (\TRACES\DHCPINI2.TXT and \TRACES\DHCPINI3.TXT) on the CD-ROM.

The following sections describe the sequence of DHCP client and DHCP server interaction. We captured network traces that illustrate these interactions in full detail.

3.1.1 DHCPDISCOVER

When a client host is started and the DHCP client is initialized for the first time, the client will broadcast a DHCPDISCOVER message on the network, sending it to UDP port 67, the BOOTP server's well-known port. The client itself uses UDP port 68, the BOOTP client's well-known port. Using these ports will ensure that a DHCP server can service both DHCP and BOOTP clients. The client is then said to be in INIT state. In Figure 57 you can see the decoded text **Port Bootp Client ---> Bootp Server**. The protocol analyzer shows the hex data 44 and hex 43 (decimal 68 and 67), which is not shown in the screen shot.

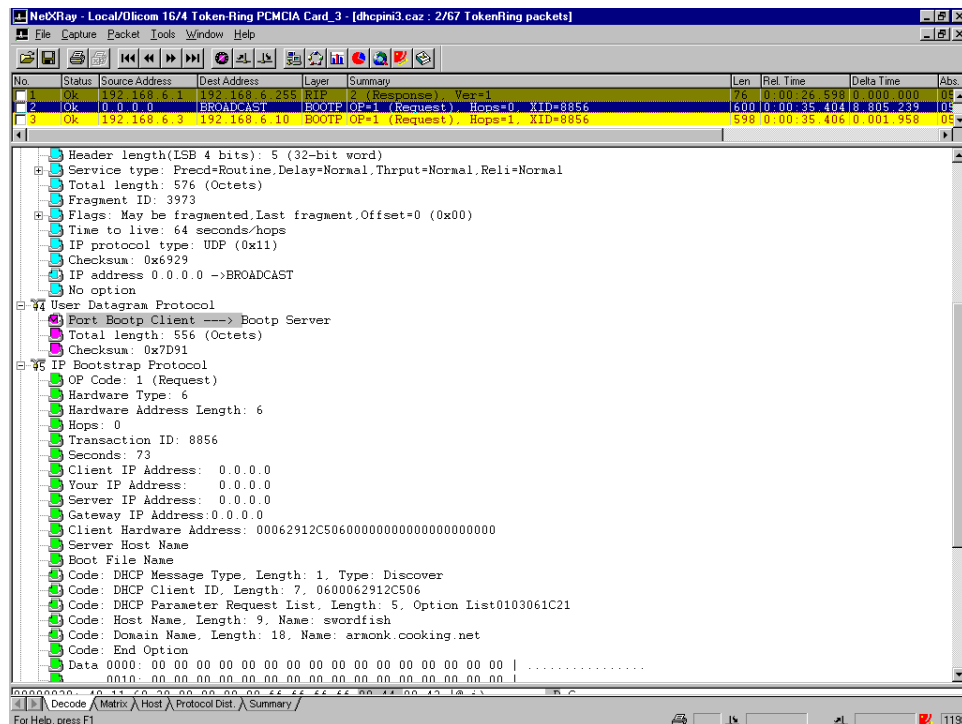


Figure 57. [(NetXRay) DHCPDISCOVER Packet

3.1.1.1 DHCP Relay Agent

If a DHCP server is not located on the same IP subnet as the client, an intermediate IP router may act as a BOOTP relay agent and forward any DHCP and BOOTP messages to a DHCP server (or to another intermediate IP router that has the same capability). In this case, the router will insert its own IP address from the subnet on which the client is located.

Software DHCP relay agents (such as DHCPRD.EXE in TCP/IP Version 4.1 for OS/2) are configured with the DHCP server addresses; so they do a directed forwarding of the frames. If the agent is used on a machine being used as a router between two subnets, the DHCP client broadcast will only occur on the subnet that the client is connected to. When the relay agent forwards the packets, they are specifically directed to the DHCP servers configured for that agent. Also, the sending and receiving ports are the same, decimal 67 (the server port), as shown in Figure 58.

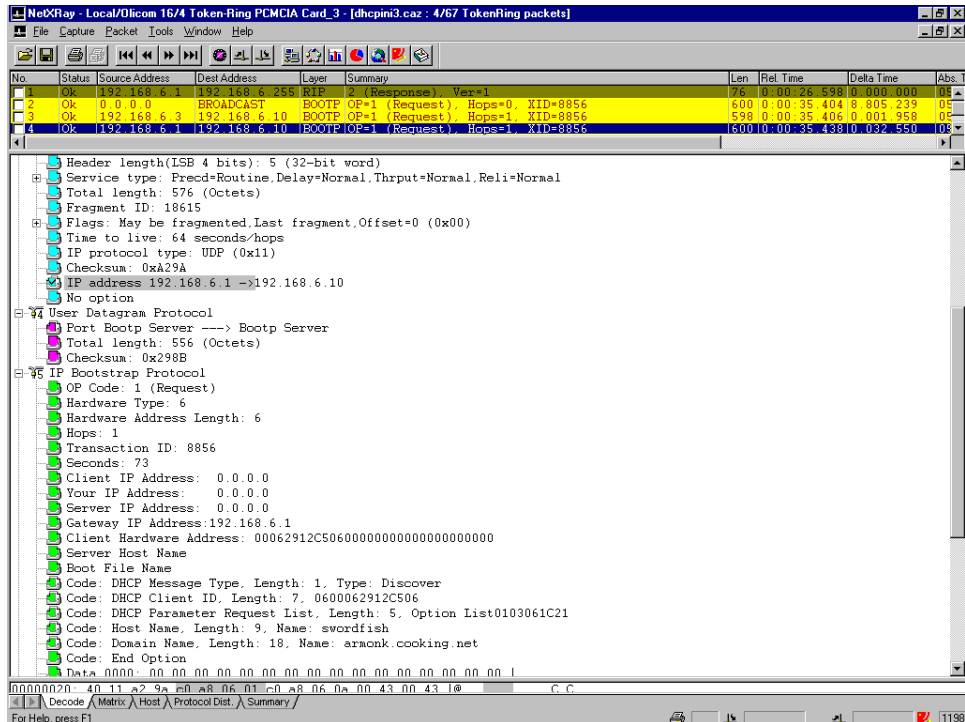


Figure 58. [NetXRay] Packets from DHCP Relay Agent to DHCP Server

Note that there are two relay agents running on the 192.168.6 subnet. You can tell because packets 3 and 4 both have a source address. Note also that the Hops count has been incremented. These relay agents did not perform any useful function for us here except for demonstration purposes.

3.1.1.2 Initial DHCP Broadcast Packets

To be able to send initial DHCP broadcast messages, a DHCP client configures its IP interface(s) with an address of 0.0.0.0 and sends the broadcast to IP address 255.255.255.255. This is shown in Figure 59 on page 78.

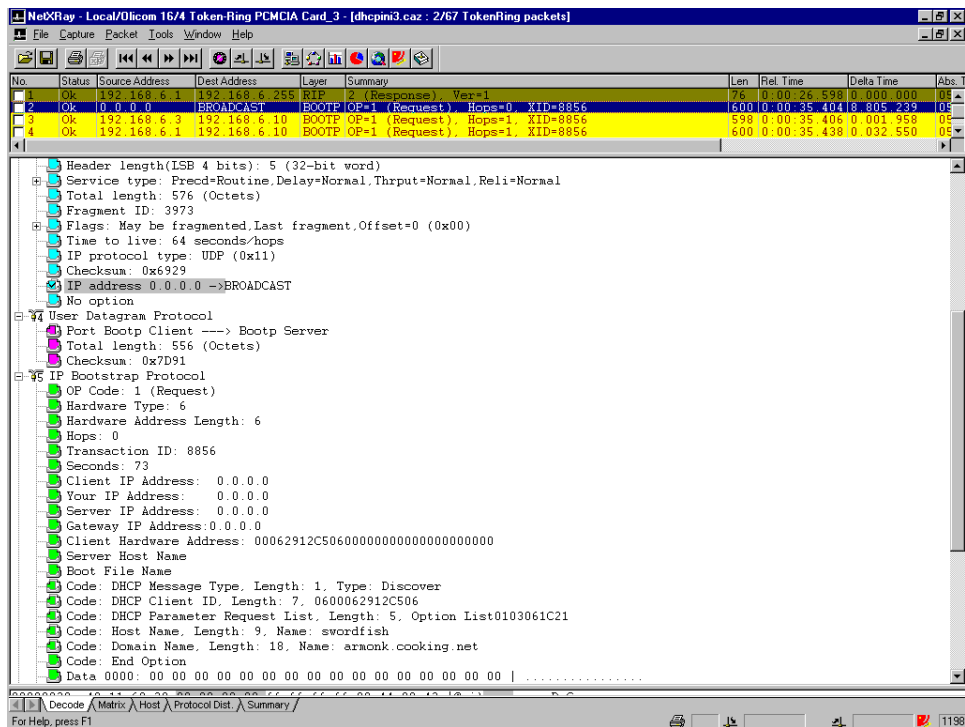


Figure 59. [NetXRay] Broadcast Packets

This is the packet that the relay agents saw and redirected using the DHCP servers address. Note that the Hops count is 0 because this is the original packet broadcast from the client.

In order to receive DHCP reply messages at a client whose IP stack has not been configured, the TCP/IP implementation at the client must be able to pass on IP packets that are sent to the client's hardware address to the IP layer in that system. Otherwise, DHCP servers (and eventually involved BOOTP relay agents) must use broadcast frames to submit their information to the client. A client will indicate its ability to receive unicast datagrams, rather than broadcast, by not setting the broadcast bit in the flags field of a DHCP message.

3.1.2 DHCPOFFER

DHCP servers that receive DHCPDISCOVER messages will respond with a DHCPOFFER message, if they have any IP addresses available. If no addresses are available at a server, it will not respond at all. A DHCP server will include an available IP address and other options in that message.

Servers may also check if an offered IP address is not already in use. They can do so using an ICMP echo request (PING) or using the Address Resolution Protocol (ARP). The TCP/IP Version 4.1 DHCP server sends an ARP packet if this is the first time the IP address is being offered out of its pool. Figure 60 shows this (we apologize that the packet numbers seem off; the numbers were changed when we filtered out some unneeded packets; we are showing everything in order, even if the packet numbering is slightly confusing).

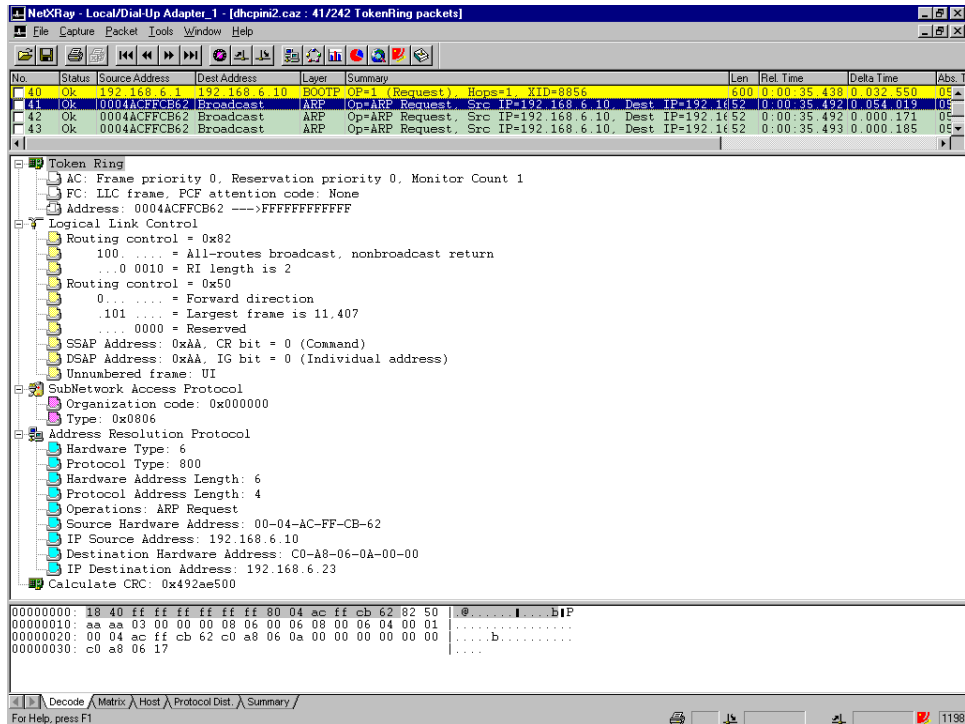


Figure 60. [NetXRay] ARP Packets

Servers may also temporarily reserve any offered IP addresses; so they will not be offered to several DHCP clients at the same time. Look at Figure 61 on page 80 to see the first DHCPOFFER that the server sent.

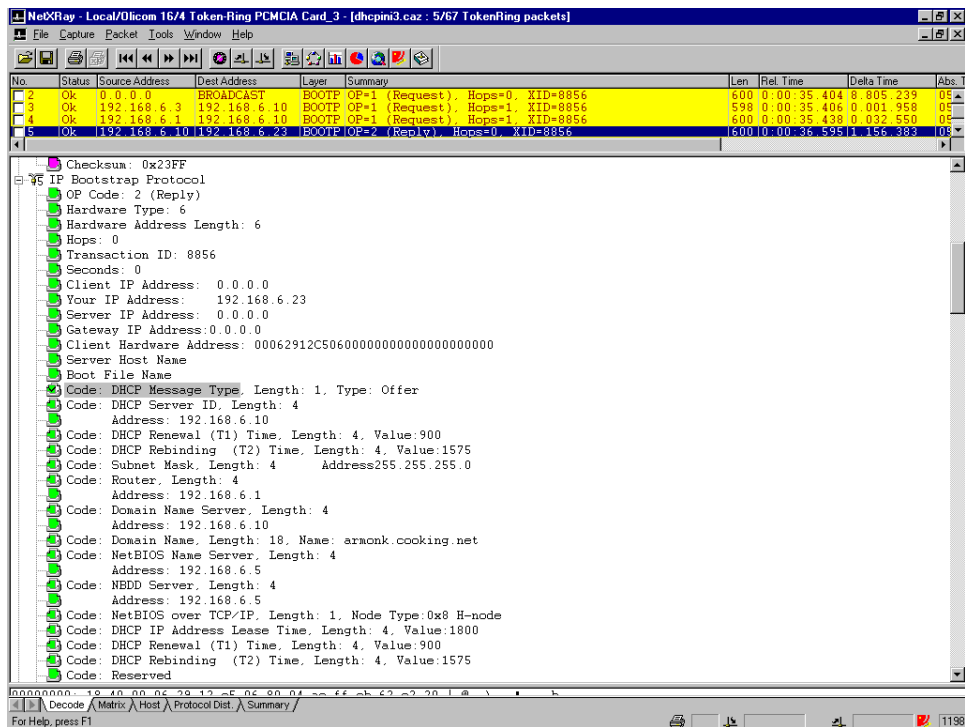


Figure 61. [NetXRay] DHCPOFFER Packet

One other item to note in this packet is decode. The last Code: that shows on the screen says *Reserved*. The actual value is decimal 201, which, at the time of this writing, Sniffer Basic did not recognize. The decoded information is incorrect and can be misleading for some or all reserved DHCP options (codes decimal 128 through 254). We hope this problem will be fixed in Sniffer Basic by the time this book is published.

3.1.3 DHCPREQUEST

A client may receive several DHCPOFFER messages from a number of DHCP servers, and it is up to the implementation of the client software to decide which server's offer the client should finally decide to accept. If a server has been selected, the client broadcasts a DHCPREQUEST message to that server whose IP address is contained in the server identifier option from the previous DHCPOFFER message.

Figure 61 shows the first DHCPOFFER packet, which was packet #5 in our trace. Packet #8 is another DHCPOFFER. Figure 62 on page 81 shows the

DHCPREQUEST packet decoded. Note that the client is requesting IP address 192.168.6.23, which is what the server offered.

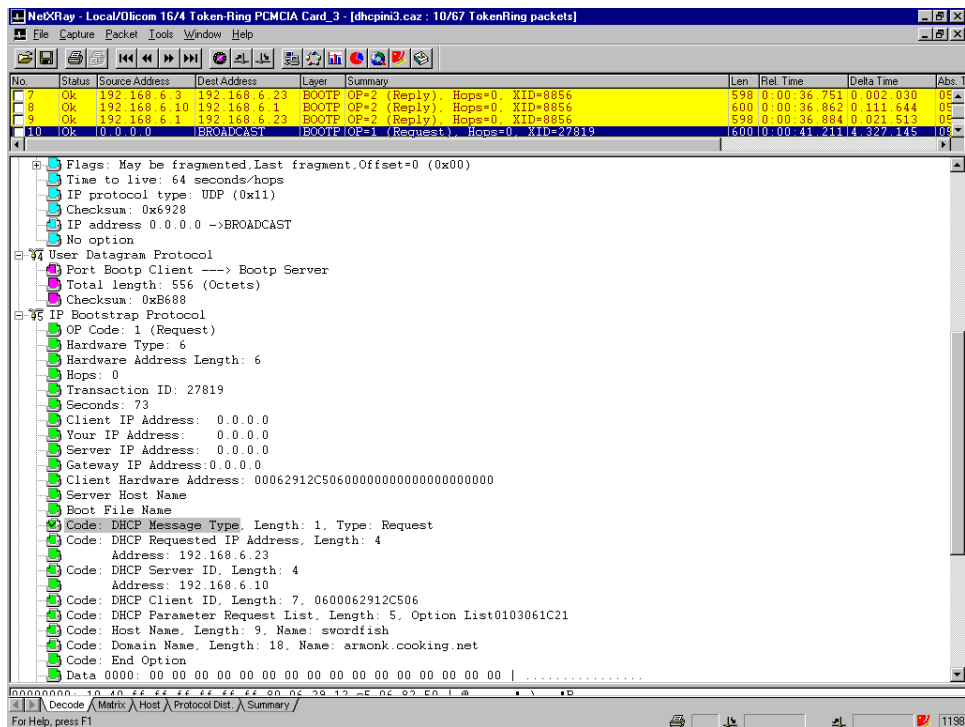


Figure 62. [NetXRay] DHCPREQUEST Packet

3.1.4 DHCPACK

The server that receives a DHCPREQUEST message from a client will finally commit the requested IP address and optimal parameters to its configuration and acknowledge that to the client by sending a DHCPACK message. If that server at that time cannot, for whatever reason, supply any of the requested configuration parameters, it will send a DHCPACK message instead. The client will then have to repeat the whole acquisition process, starting with a DHCPDISCOVER message. Figure 63 on page 82 shows the servers acknowledgment packet.

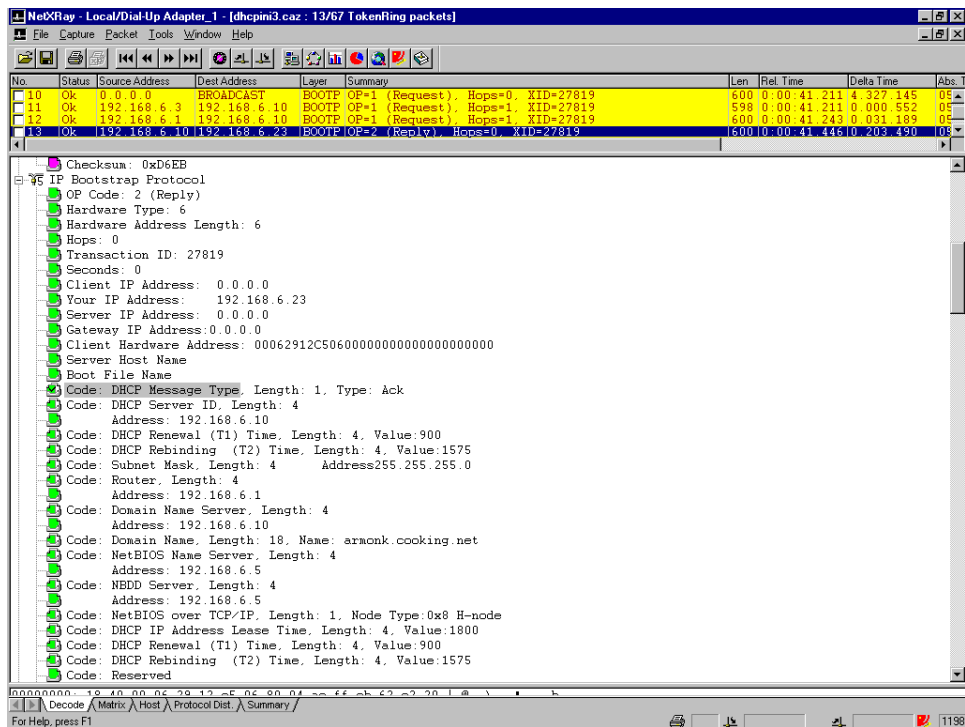


Figure 63. [NetXRay] DHCPACK Packet

3.1.5 Verification Process

After receiving DHCPACK, the client should also check to be sure the offered IP address is not already in use. This can be done using ARP rather than PING since, at that time, the client has no IP host address it can use.

If the offered address is already in use, the client responds with a DHCPDECLINE message to the server; otherwise, it will configure its IP interface(s) according to the values obtained from the DHCP server. The client is now fully configured, which is also referred to as the BOUND state. Figure 64 on page 83 shows the client checking the network for the IP address (again, please bear with us on the packet numbering).

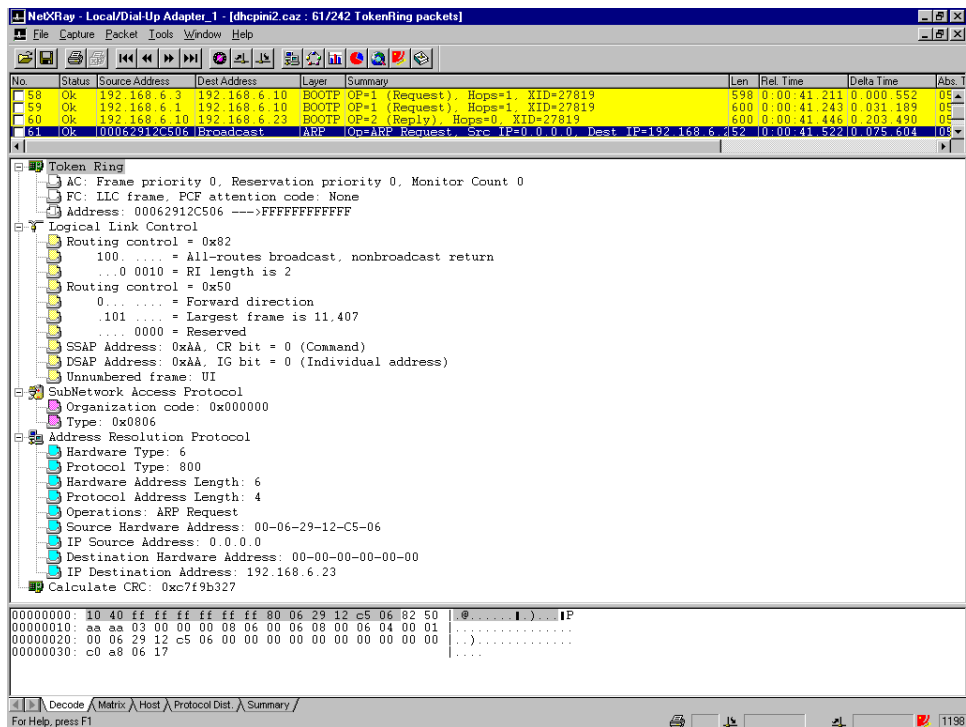


Figure 64. [NetXRay] Verify State (ARP)

After sending a DHCPDECLINE message, the client must restart the entire acquisition process, starting with a DHCPDISCOVER message. The server, in this case, must mark that address as not available, and it may notify the administrator with an information message.

If a client does not receive any DHCPPOFFER messages, it will continue to broadcast DHCPDISCOVER messages at random intervals for a certain period of time, before it will notify the user with an error message that it could not obtain any TCP/IP configuration parameters.

3.1.6 DHCPRELEASE

When a client no longer needs a provided TCP/IP configuration, it may inform the server about that using a DHCPRELEASE message. The server will then mark the IP address as available. This message will not be acknowledged by the server. We show the DHCPRELEASE packet in Figure 65 on page 84.

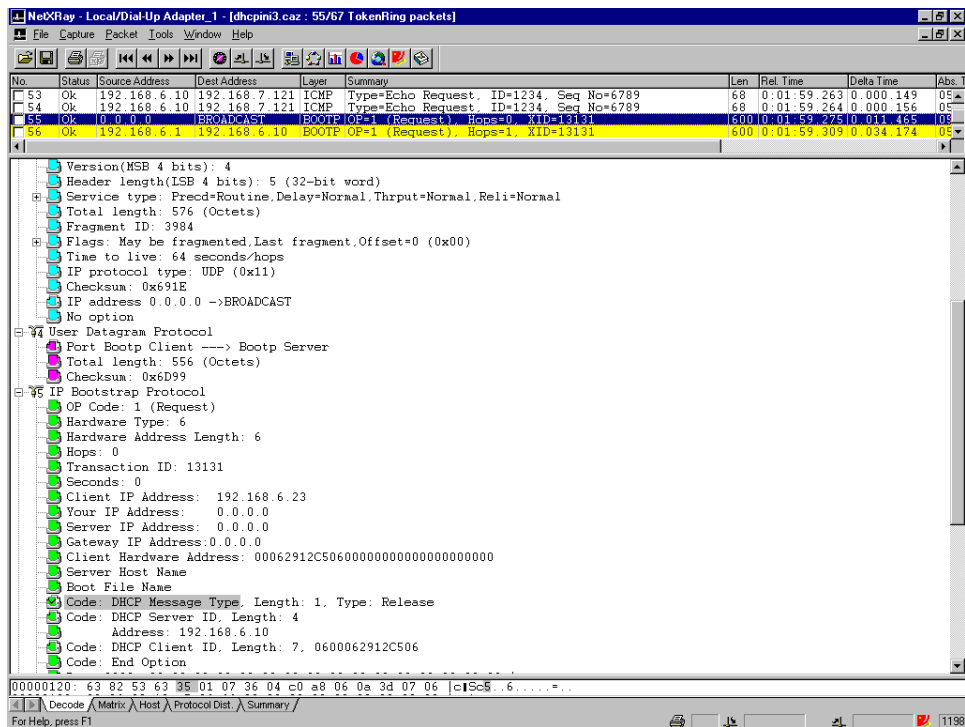


Figure 65. [NetXRay] DHCPRELEASE Packet

Note that even though the client has the leasing server IP address, the client still broadcasts the release packet to help assure that the server receives the packet.

3.2 DHCP Renewing, Rebinding and Rebooting Processes

This section describes the interaction between DHCP servers and clients that have already been configured. If a client uses multiple IP interfaces, each of them must be configured separately by DHCP. The following describes steps for rebinding and rebooting:

1. After a DHCP client has applied the TCP/IP configuration parameters, which it has obtained from a DHCP server, it also received a lease time during which the client is rightfully entitled to use the given configuration. Two timers, T1 and T2, will start to tick down. While T1 will expire before T2, T2 will expire before the end of the assigned lease time. According to the latest IETF Internet draft, T1 defaults to 0.5 times of a lease time, and

T2 defaults to 0.875 times of a lease time, but either timer can be set by the server through DHCP options.

2. When timer T1 expires, the client will send a DHCPREQUEST message to the server asking to extend the lease for the given configuration. This state of a client is called the RENEWING state. The server would usually respond with a DHCPACK message indicating the new lease time to which T1 and T2 will then be reset accordingly.
3. If no DHCPACK is received until timer T2 expires, the client enters the REBINDING state. It now has to broadcast a DHCPREQUEST message to extend its lease. This request can be confirmed by a DHCPACK message from any DHCP server on the network.
4. If the client does not receive a DHCPACK message after its lease has expired, it has to stop using its current TCP/IP configuration and may start over from the INIT state as described earlier.
5. If a client has been configured before and is rebooted, it may want to use the previous configuration values, which may have been stored in a file on the client's hard disk. In that case, the client would broadcast a DHCPREQUEST message containing the desired parameters in the appropriate option fields. DHCP servers will respond with DHCPACK messages if they can supply the requested configuration. If no DHCPACK messages are received by the client, it may wait and then start over from INIT state, as described earlier.
6. If a client is using external configuration values (external to DHCP), which it may have obtained through manual configuration, it would assemble a DHCPINFORM message containing its current configuration and any additionally desired parameters. If the client knows a DHCP server's IP address, it will send this message to that address; otherwise, it will broadcast the message. A server will respond to that request using a DHCPACK message which only contains the additionally required options for the client. If the client does not receive any replies, it should notify the user of that problem and continue operation using suitable defaults.

Chapter 4. Serving Names

This chapter describes how to assign unique names, statically and dynamically, to all devices on your TCP/IP network. We will see how a dynamic name server works and how it integrates with existing static name servers and firewalls. We will also discuss how to choose a naming scheme, and whether your systems should have (or even need) a hostname.

4.1 Why Names?

Why would you want a DNS server, anyway (much less a dynamic one)? You do not really have to have one, but consider the following scenario.

For this scenario, we will assume that you are not even using a DHCP server, that all IP addresses are hard coded.

You have your network of, say 20 (or even just 10) machines and a server. You have just configured all of the names in all of the host files of all of the machines. Now, you can reach Mary's machine without having to remember, for example, 192.168.20.32. A week later, Bob is hired, and you configure another machine for him. Now, if you want all the other machines to be able to reach Bob without having to use the IP address, you will have to go to each and every machine and update the host files. Of course, you could be clever and download new files to each one and the associated verifications that would involve.

Let's add a DNS server (also known as a *name server*) to the preceding scenario. Still, we will say there is no DHCP server, you are hard coding addresses. All the machines are configured to resolve names to addresses by making requests to the name server (this is, of course, transparent to the user). You add Bob's machine to the network. You make some additions to the name server configuration files, and now, all the machines know how to reach Bob. Bob knows how to reach all the machines.

Now that we have a DHCP server in the network (setting up a DHCP server is covered in Chapter 2, "Up and Running with DHCP" on page 17), when you plug Bob's machine into the network, you do not even have to assign an address to his machine. But we also have a Dynamic Domain Name System (DDNS) server. Now, it is as easy as plugging Bob's machine into the network. That's it. The address assignment is automatic, and the name server update is automatic (Dynamic, right?). All machines can reach Bob, and Bob can reach all machines.

4.2 What is DNS?

The previous section explains what names can do for you, but just what is a Domain Name System? A Domain Name System maps host IP addresses to hostnames. Just as you may have told your mom when you were young "I'm going over to Mary's to play," rather than "I'm going over to 1141 South Emerson Street to play," DNS servers allow you to reach another host from your host without having to provide the explicit address. DNS servers are constantly updated by system administrators and by other DNS servers.

Before DNS, the only way to get name/address mappings updated was from a single master list. In the earliest (1970s and early 1980s) days of the Internet, there was a single definitive source for the Internet hosts file. All other servers received the name/address mappings from a special file, the HOSTS.TXT file. When a host wanted to update its host file, it would connect to the definitive host and FTP the new file. There were fewer hosts on the Internet then. (This was back when the Internet was actually called ARPANET and was run primarily by the Department of Defense.) This system was not practical because the number of network hosts started growing. Hence, the development of the Domain Name System.

The DNS is a program consisting of a database of name and address information. The program responds to queries from other programs (called *resolvers* or *stub resolvers*). DNS servers can communicate amongst themselves to update each other with new information. This mechanism is what allows you to reach a Web page on a new server when a friend sends you a link. You do not have to know the IP address of the server that contains the Web page. Chances are that your own Internet service provider's name server may not contain the information you want, and so it queries another name server. The name servers cooperate in order to keep the information current. At least one of the DNS servers need someone to manually update the information on them.

The structure of the DNS database is similar to the structure of a file system. The whole database or file system is pictured as an inverted tree with the root system at the top. Each node in the tree represents a partition of the database. Each domain or directory can be further divided into partitions, called subdomains (such as the file system's subdirectories).

The domain name space is *tree* structured. The top-level domains divided the Internet domain name space organizationally. Examples of top-level domains are:

- *com*: Commercial organizations, such as IBM (ibm.com), CNN (cnn.com), and mycompany (*mycompany.com*). ibm is a subdomain of the top-level domain *com*.
- *edu*: Educational organizations, such as the University of Minnesota (umn.edu), New York University (nyu.edu).
- *gov*: Government organizations, such as the Federal Bureau of Investigation (fbi.gov) and the National Science Foundation (nsf.gov).

The tree is limited to 127 levels; this is a limit on subdomains, although there is no limit on the number of branches at each node.

Each node in the tree is labeled with a name (see Figure 66). The root has a null label (" "). The full domain name of any node in the tree is the sequence of names on the path, from the node up to the root, with a dot between node names. For example, in Figure 66, if you follow the arrows from the bottom label to the top, from the host: *www* to the root label, you can form the full domain name for that host: *www.as400.ibm.com*.

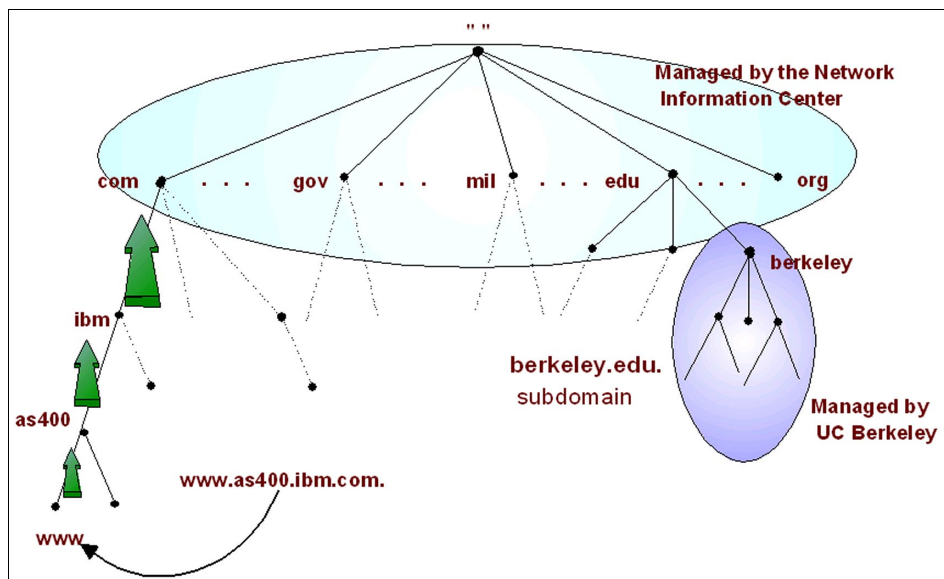


Figure 66. DNS Name Space

In DNS, each domain can be administered by a different organization. Each organization can then break its domains into a number of subdomains and dole out the responsibility for those domains to other organizations. This is because DNS uses a distributed database, where you can manage your own

domain (company.com), or parts of the name space (subdomains) can be delegated to other servers (department.company.com).

The DNS servers responsible for the top level Internet domains, such as com, are also called *Internet root servers*, and they manage information about the top-level domains. For example, the Internet's Network Information Center runs the edu domain, but assigns U.C. Berkeley authority over the berkeley.edu subdomain.

Domains can contain both hosts and other domains (their subdomains). For example, the ibm.com domain contains hosts, such as www.ibm.com, but it also contains subdomains such as as400.ibm.com.

Domain names are used as indexes into the DNS database.

Each host on a network has a domain name with a DNS server that points to information about the host. This information may include an IP address, information about mail routing, and so on.

Why this complicated structure? It is to solve the problems that a host table has. For example, making names hierarchical eliminates the problem of name collisions. Domains are given unique domain names; so organizations are free to choose names within their domains. Whatever name they choose, it does not conflict with other domain names, since it has its own unique domain name.

For example, we can have several hosts named www, such as www.ibm.com and www.yahoo.com, because they are in different domains managed by different organizations. See Figure 67 on page 91.

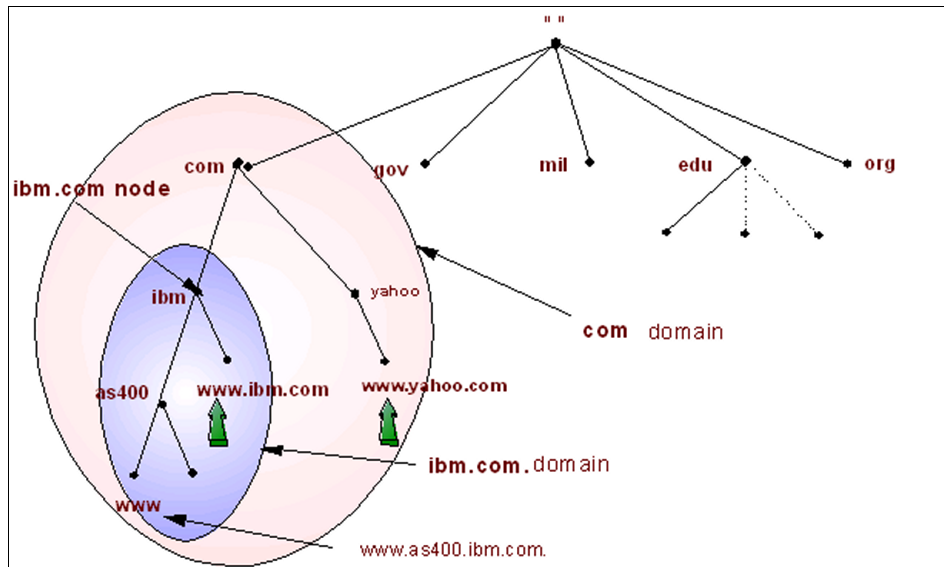


Figure 67. Hosts with the Same Names in Different Domains

We can have a host in the same domain that also has the same hostname, such as `www.ibm.com` and `www.as400.ibm.com`, because they belong to different subdomains.

4.3 Domain versus Zone of Authority

The concept of domains versus zones of authority can be a confusing one. We do our best to explain it in this section.

One of the main goals of the design of the Domain Name System is decentralization. This is achieved through *delegation*. The central DNS administrator in your company administering the company's domain can divide it into subdomains. Each subdomain can be delegated to other administrators. This means that the administrator delegated to becomes responsible for maintaining the subdomain.

Domain versus Subdomain

A *domain* is a subset or subtree of the name space tree. A *subdomain* is a subset of the domain.

Figure 68 on page 94 shows the domain *mycompany.com* as a subset of the *.com* name space. Under *mycompany.com*, there are other subdomains, such as:

- *endicott.mycompany.com*
- *rochester.mycompany.com*
- *otherdomain.mycompany.com*

Name servers are programs running on a system, such as OS/2 Warp Server, AS/400, or AIX, with DNS support. In Figure 68 on page 94, the following hosts are running name server programs:

- *as1.mycompany.com*
- *rst.rochester.mycompany.com*
- *otherhost.otherdomain.mycompany.com*

They are called Domain Name System (DNS) servers or simply name servers.

Name servers have information about the part of the domain name space called a *zone* or *zone of authority*. Both domains and zones are subsets of the domain name space. A zone contains host information and the same data that the domain contains, excluding the information that is delegated somewhere else. If a subdomain of a domain is not delegated, the zone contains host information and data for the subdomain.

Name servers have complete host information and data for a specific zone. Name servers are said to be *authoritative* for the zone for which they have this complete host information and data.

As shown in Figure 68 on page 94, the *mycompany.com* domain is divided into the following subdomains:

- *endicott.mycompany.com*
- *rochester.mycompany.com*
- *otherdomain.mycompany.com*

The zone *mycompany.com* contains the following hosts:

- *as1.mycompany.com*
- *as2.mycompany.com*
- *as5.mycompany.com*
- *NTserver1.mycompany.com*

It also contains the host information and data in the subdomain *endicott.mycompany.com*:

- *host1.endicott.mycompany.com*
- *host2.endicott.mycompany.com*

The subdomain *endicott.mycompany.com* has not been delegated, and its host information and data remain in the *mycompany.com* zone. The administration of the *endicott.mycompany.com* is the responsibility of the *mycompany.com* administrator. *AS1.mycompany.com* is the name server that has complete host information and data for the *mycompany.com* zone of authority.

The zone *mycompany.com* does *not* contain information in the subdomains that have been delegated.

A subdomain of *mycompany.com* is *rochester.mycompany.com*, and its administration has been delegated. The zone *rochester.mycompany.com* includes host information and data in the following subdomain:

- *rochester.mycompany.com*

which includes host information and data in these subdomains:

- *rst.rochester.mycompany.com*
- *host1.rochester.mycompany.com*
- *host2.rochester.mycompany.com*

The DNS server that has complete host information and data for the *rochester.mycompany.com* zone is *rst.rochester.mycompany.com*.

A subdomain of *mycompany.com* is *otherdomain.mycompany.com*, and its administration has been delegated. The zone *otherdomain.mycompany.com* includes host information and data in the subdomain *otherdomain.mycompany.com*:

- *otherhost.otherdomain.mycompany.com*
- *otherprinter.otherdomain.mycompany.com*
- *otherserver.otherdomain.mycompany.com*

The DNS server that has complete host information and data for the *otherdomain.mycompany.com* zone is *otherhost.otherdomain.mycompany.com*.

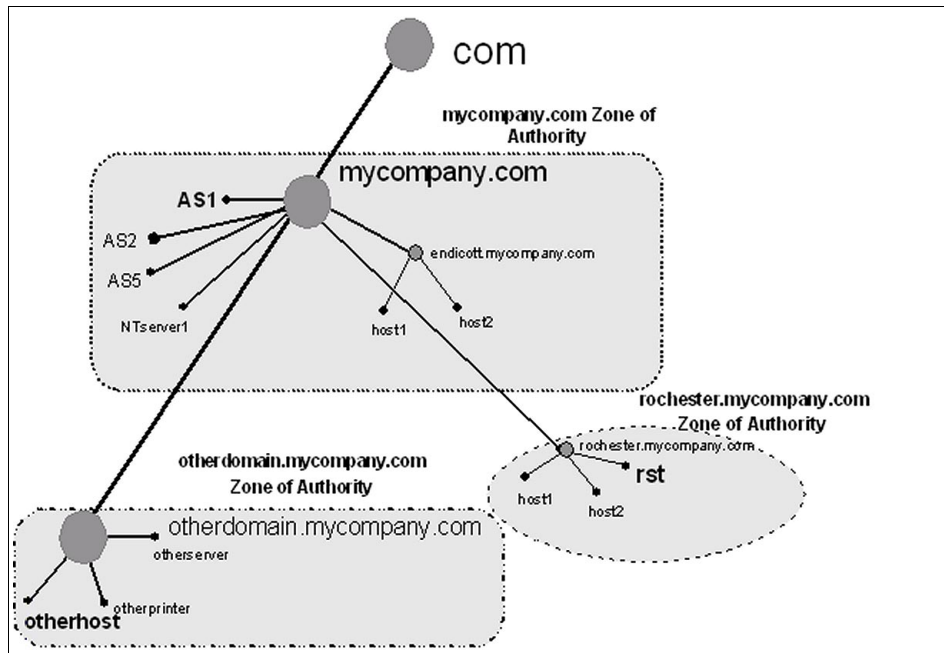


Figure 68. Domain, Subdomain, Delegation, and Zone of Authority

4.4 Differentiating Name Servers

Name servers are devices that store the information about the domain name space. Usually, they have complete information about some part of the name space or zone. There are several types of name servers, and they are illustrated in the following sections.

4.4.1 Static Name Servers

A static name server requires someone to manually edit and update the lookup table, whenever an IP address gets assigned or reassigned to a particular name, or when a name is no longer required because the address or name are no longer required. When changes are made to the lookup table, the static DNS server needs to be restarted to make the changes active.

4.4.2 Dynamic Name Servers

A dynamic name server is capable of updating the lookup table itself whenever a DDNS client or DHCP server informs the DDNS server to update a client's hostname with a certain IP address that was assigned by a DHCP server. A dynamic name server never needs to be restarted. IBM's DDNS

servers on AIX and OS/2 Warp Server (and TCP/IP Version 4.1 for OS/2) can be used as static DNS servers also. Refer to “The Dynamic Domain Name System (DDNS)” on page 112, for more detailed information.

4.4.3 Primary Name Servers

This is the server that the hosts in the zone of authority are configured on. It is the server that the DNS administrator configures and maintains. When this server gives responses to queries from its primary domain files, the responses are called authoritative. A name server for a primary domain reads the primary domain configuration information directly from files configured by DNS administrator and/or updated by dynamic clients.

4.4.4 Secondary Name Servers

This server has the same information as the primary name server. However, instead of getting its information directly from the DNS administrator configuring it, it gets its information from another name server through zone transfers over the network. Secondary name servers are authoritative servers, just like the primary name servers, and they act just like the primaries in terms of data storage and data queries.

Primary vs. Secondary Name Server

A DNS server can be a primary name server for one or more domains, as well as a secondary name server for one or more domains. The terms *primary name server* and *secondary name server* are somewhat misnomers, because any given name server may be primary from some zones and secondary for other zones.

A *zone transfer* is a TCP/IP transfer of domain files from another DNS server (called a master name server). This is done automatically when the secondary name server starts and also when the secondary name server detects that its domain files are downlevel from the master name server's domain files. The zone transfer is initiated from the secondary name server. The zone transfer cannot take place if the master name server is not active.

A secondary name server is used for two reasons: spreading the DNS query workload over more than one server and as a backup in case the primary name server stops responding.

The name servers configured at the client very likely have no direct relationship to the zone being queried, and even if they do, the name servers might all be secondaries, or the first defined name server might be a

secondary. Don't be confused with the terms *primary* and *secondary* name server at the client which applies when a client is configured with more than one DNS server for name resolving. If the first name server (also called the primary name server) does not respond, the client can query the second name server (also called the secondary name server). When the secondary name server gives out a response to a query, the response is also called authoritative. In other words, an answer from a secondary name server is considered to be just as good as if the answer came from a primary name server. Same applies if there is a third name server (also called tertiary name server) defined at the client resolver configuration.

4.4.5 Master Name Servers

This is the name server from which a secondary name server gets its zone transfer from. A master name server can either be a primary name server or another secondary name server.

4.4.6 Caching-Only Name Servers

A name server that does not have authority over any zone is called a caching-only name server. It gets all of its information by querying. A caching-only name servers's responses are always non-authoritative.

4.4.7 Authoritative Name Servers

A server that is considered to be authoritative for a domain is either the primary server or a secondary server for that domain. If another name server or a client queries either the primary or the secondary name server for information that they are authoritative for, the response is considered to be authoritative. Can a name server that is not authoritative over a domain give a response to a client about that domain and have that response considered an authoritative response? The answer is yes. If the non-authoritative server does not know the answer and queries an authoritative name server on behalf of the client and then returns the answer to the client, this response is considered to be authoritative. The non-authoritative name server caches this information. If a second client requests this same information from the non-authoritative name server (and this information is still in its cache), the name server gives the response to the client, but now, this same information is labeled non-authoritative. Why? Because the information in the response this second time came out of the name server's cache. Another way of saying this is: at some point, a non-authoritative response came out of a name server's cache.

4.4.8 Parent and Child Name Servers

The concept of parent and child domains is equivalent to the concept of domain and subdomain: once your domain grows to a certain size, you may need to distribute management by delegating authority of part of your domain to one or multiple subdomains. The upper-level domain is the parent and its subdomains the children.

The name server authoritative for the parent domain is the parent name server, and the one authoritative for the subdomain is the child name server. For example, in Figure 68 on page 94, *otherdomain* is a subdomain of *mycompany.com*. If a DNS server, AS1, is configured to be responsible for the *mycompany.com* zone of authority, and the authority for the zone *otherdomain.mycompany.com* is delegated to another DNS server, *otherhost*, then AS1 is considered to be the parent name server and *otherhost* is considered to be the child name server.

4.4.9 Root Name Servers

Internet root name servers know where name servers that are authoritative for the top-level domains are, and most of the Internet root name servers are authoritative for the top-level organizational domains (.com, .edu, .net, and so on). The top-level domain servers have information about the second-level domain that a given domain is in.

A company can implement internal root name servers. In this case, given a query for a company's subdomain, the internal root name server can provide information for the second-level subdomain the queried subdomain is in.

A root name server is configured in a lower level name server to help it to navigate the name space tree top down, when it cannot answer a query with authoritative data or data in its cache.

If we use the example discussed in the previous section, the DNS server *otherhost* is authoritative for the zone *otherdomain.mycompany.com*, shown in Figure 68 on page 94. The AS1 name server is authoritative for the *mycompany.com* zone of authority *and* is configured as the internal root for the whole company's name space. The internal roots can run on host systems all by themselves, or a given host can perform double duty as an internal root and as an authoritative name server for other zones. If *otherhost* cannot answer a query, it asks its root name server, which is AS1, the DNS server at the top of the *internal* name space tree. We stress *internal*, because in this example, these DNS servers are only part of an internal network. We are assuming that the network does not have Internet access; thus, the Internet com node is not part of this DNS name space tree. Therefore, the

DNS server AS1 in the domain *mycompany.com* is at the top of tree. A root name server can be thought of as the name server at the *top* of the DNS name space tree. Remember, the DNS name space tree may be different, depending on whether the network is an internal network, or if the network includes the Internet DNS name space.

4.4.10 Forwarders

A DNS server can be configured to send queries to which it does not know the answer to a DNS server called a forwarder name server. Whereas going to a root name server for help in answering a query can be thought of as going to the top of the DNS name space tree, going to a forwarder can be thought of as going sideways in the DNS name space tree for help. The DNS administrator configures which DNS server is the forwarder. Usually, several DNS servers are configured to have the same forwarder. Then, the forwarder name server is configured with the root name servers (for example, the Internet root name servers). If the forwarders cannot answer the query, they query the root name servers, get the answer, and cache it. This way, a forwarder name server can build up a large cache of information. As the cache increases, chances are that the forwarder will receive a query for which it has a cached answer. This, in turn, reduces the number of times a root name server needs to be queried. Using a forwarder name server is an opportunity to build a large cache of information on one (or just a few) name servers.

4.4.11 Firewall Name Servers

A firewall name server is a special application of forwarding name servers. This could be useful when you want to connect a private network (intranet) to the Internet. You usually do not want all of your private DNS server information available to the Internet. You provide a more limited content DNS server to the public Internet (your Firewall DNS server). When a host on your private network requests an address, it first queries the private DNS server. If that name server does not have the information desired, it sends the request to the firewall name server which forwards the request out to the public Internet.

Please refer to the Chapter 8, “Security of DHCP and Dynamic DNS” on page 309, to read how to connect your private network (intranet) to an untrusted network. This could be a customer’s intranet or the global Internet. In either case, what we show you will help protect your internal resources from the malicious or just plain curious folks out there.

4.4.12 Record Types

Name server records are called resource records (RR) and are divided into classes for different kinds of networks. We will only talk of the Internet class of records here. The IN that you see as the second field in most records means Internet. This list is not complete; see RFC 1035 for all RR types.

Figure 76 on page 110, Figure 77 on page 111, Figure 99 and Figure 100 on page 146, show examples of files that contain each record; so you can see the format of each one. These records are:

- **SOA** records are start of authority records, indicating that this name server is the main source of authoritative information for this domain. There is only one SOA record in the configuration files, and it is required to be in the zone and the address files.
- **NS** records are name server records. Each NS record should indicate a valid name server. Sometimes, this is not the case when you receive information from other name servers. As an aside, if you find a name server you know to provide bad information (or no information), you can use the `bogusns` directive in your boot file. For example, `bogusns 192.168.6.11` would tell your name server not to query this name server (hopefully, there are other name servers that will be able to provide valid information for you).
- **RP** records are the responsible persons for the domain. If you found a name server that was delivering bad (or no) information, you would probably want to try to contact the RP for that name server and rectify the problem prior to adding a `bogusns` record to your name server.
- **TXT** records in the DDNS database may provide more information about how to contact the RP. A text record can contain arbitrary information in quotes (for example, a telephone number and name). It basically maps any kind of information to a hostname. This is a feature that was created as a tool for the system administrator to map a hostname and/or IP address lease to a real person. This is done by setting labels in the DHCP server in option 192 (supported by TCP/IP Version 4.1 for OS/2 DHCP server).
- **MX** records are mail exchanger records. An MX record indicates a host machine that will either deliver mail to the addressee, or forward (using SMTP) the mail to another host that is closer to the addressee. The other host will either deliver or forward, as necessary. MX records have a third parameter to prevent mail loops, the preference value. The preference value is a relative value for each mail host. The lowest value would indicate the highest preference for delivery (that is, a mailer should

forward undeliverable mail to that host first). A preference value can be between 0 and 65535 (a 16 bit unsigned value).

- **A** records are address records. This is a record in the [D]DNS that maps a hostname to an IP address so that you can resolve an IP address using by issuing a name. See also PTR records.
- **RR** entries are resource records in the nameserver database. A resource record maps information to a hostname.
- **PTR** are pointer records. This is a record in the [D]DNS database that maps an IP address to a hostname so that you can resolve a hostname using its IP address. See also A records.
- **CNAME** records map a host alias name to the canonical name of the host (that is, the fully qualified domain name of the host that is defined by an address record).
- **HINFO** records contain host information and are generally not recommended to use since you do not want to provide information that might be useful for any potential hackers. However, TCP/IP Version 4.1 for OS/2 makes use of HINFO for ProxyArc clients. The ProxyArc client's MAC address is stored here in an encoded fashion to ensure that other clients cannot take over other clients' hostnames.
- **KEY** entries in the DDNS database contain all hostname/domain name/primary name server settings that have been created and the associated keys to be able to update those hostnames.

4.5 Windows NT as a Static DNS Server

This section shows you how to set up your Windows NT server as a static DNS server using an example of a simple internal network environment. Note that Windows NT's implementation of DNS *cannot* be configured as a dynamic DNS server.

4.5.1 Scenario

In this section, we use a simple network environment with only one subnet. This network is not connected to the Internet or any other networks. In this network, we assume that we have a DHCP server on the Windows NT server up and running. We also have DHCP client computers. In this network, we initially do not have a DNS server.

We configure the Windows NT server as a static DNS server. We also configure Windows NT server as a WINS server later in this chapter.

In our environment, our domain name is bellevue.cooking.net, and the DHCP/DNS server's hostname is mustard. Table 4 illustrates our network configuration:

Table 4. Windows NT Server Configuration Information

Configuration Settings	Value
Network ID	192.168.8.0
Subnetmask	255.255.255.0
DHCP/DNS/WINS Server IP address	192.168.8.10
DHCP/DNS/WINS Server Hostname	mustard

4.5.2 Tasks

The tasks to complete the scenario are as follows, and the sections following this describe the steps of each task:

1. Plan the primary domain.
2. Install the DNS server service on the Windows NT server.
3. Configure DNS search order on the Windows NT machine.
4. Create a DNS primary name server using the DNS manager.
5. Create a new zone for the forward mapping.
6. Create a new zone for the reverse mapping.
7. Configure hosts.
8. Add DHCP options for a DNS server.
9. Enable WINS lookup, if desired, and add DHCP options for the WINS server.
10. Start the DNS server.

4.5.3 Planning

The first step is to design your domain. Before you establish domain name servers in your network, you need to determine a domain name for your domain. You also need to determine the server that is going to be the primary DNS server for your domain. In our scenario, bellevue.cooking.net is the primary domain on the Mustard name server. All the hosts on the 19.168.8.0 network are included in the domain bellevue.cooking.net.

4.5.4 Setting Up

4.5.4.1 Installing the DNS Server

The default installation of Windows NT Server 4.0 does not install the DNS server service. To install the DNS server service, follow the steps below:

1. Open the **Control Panel**.
2. Double-click the **Network** icon.
3. On the Network notebook, click on the **Services** tab and click on the **Add...** button.
4. Select **Microsoft DNS Server** from the list and then click **OK**.
5. When prompted, enter the path to the Windows NT files (Windows NT 4.0 Server CD) and click **Continue** to copy the files and finish the installation.
6. After installation, restart your computer.

4.5.4.2 Verifying TCP/IP Configuration

The following steps describe how to set up your Windows NT computer to run the DNS server. All the values we used to configure in the following steps are the values we defined in our sample network.

Before you start configuring, make sure that TCP/IP is configured correctly. The DNS server must have a static IP address, and therefore, cannot be a DHCP client.

The DNS server computer itself can also be a DNS client. Follow the steps below to add the DNS server information:

1. Open Control Panel and then open the **Network** icon.
2. Click on the **Protocols** tab.
3. Select **TCP/IP Protocol** and click on **Properties**.
4. Click on the **DNS** tab and type a hostname and domain name.
5. Under DNS Service Search Order, click **Add**.
6. In the DNS server box, type the IP address for your computer (DNS server) and click **Add**.
7. Click **OK** to close the notebook.
8. Click **OK** to close the Network notebook.

4.5.4.3 Configuring DNS Server

The following steps illustrate how to set up a static DNS server:

1. By default, the Microsoft DNS server service starts automatically when the computer is started. If the DHCP server is not running, open the **Service** icon in the Control Panel, select the **Microsoft DNS Server** in the service list and click on the **Start** button. You can also start the DNS server at the command prompt by using the following command:

```
NET START DNS
```

2. Open the DNS manager. To do so, select [**Start — Programs — Administrative Tools (Common) — DNS Manager**].
3. Click the DNS menu and select **New Server**.
4. Enter the hostname of the DNS server and click **OK**.



Figure 69. [Windows NT] Adding DNS Server

5. Next, you need to add a new zone for the forward mappings. In the Server list, highlight the DNS server you have just created and select **New Zone** from the DNS menu.
6. Check **Primary** for the zone type and click **Next>**.
7. Enter your full domain name for the zone name. Then, move your cursor to the space for the file name. **<ZONE_NAME>**. DNS is the default name for the zone file. If you want a different name, change the file name and click **Next>**.

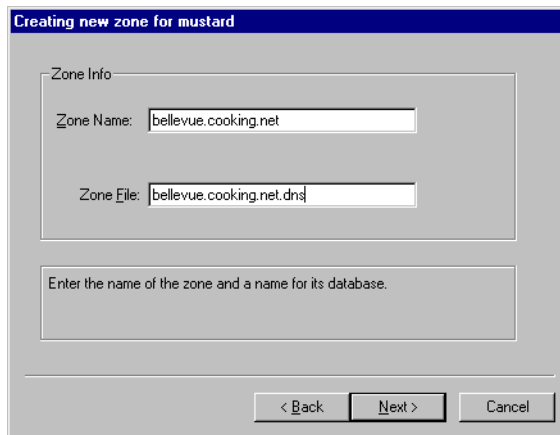


Figure 70. [Windows NT] Creating a New Zone for Forward Mappings

8. Click **Finish** to create the zone.
9. If you want to have reverse lookups, you need to create a zone for the reverse mappings. Repeat steps, five through eight above. This time, the naming is different. For example, if a network ID is 192.168.8.0, a zone name for the reverse domain would be 8.168.192.in-addr.arpa.

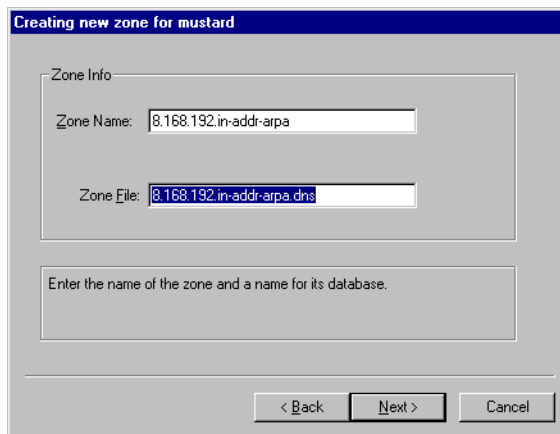


Figure 71. [Windows NT] Creating a New Zone for Reverse Mappings

10. Select **Update Server Data Files** from the DNS menu to affect the changes immediately to the zone files.

4.5.4.4 Adding Hosts

The next step is to add hosts to the domain.

1. Highlight the zone for the forward mappings and select **New Host** from the DNS menu.
2. Enter the hostname and IP address. If you made a zone for reverse mappings, check **Create Associated PTR Record**, then click **Add Host**.

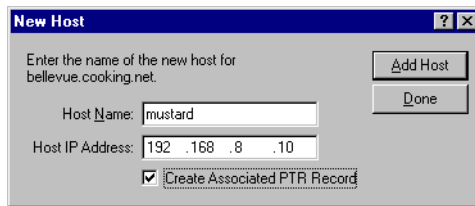


Figure 72. [Windows NT] Adding a New Host

3. Repeat steps one and two above for all the hosts in your network.
4. When you finish adding hosts, click **Done**.

4.5.4.5 Adding DHCP Options for DNS Server

You need to configure the DHCP server so that DHCP clients can get the configuration information for the DNS server and domain name. In our scenario, we added following 2 options:

- **Option 006 DNS servers:** IP address of the DNS server
- **Option 015 Domain name:** Domain name

Refer to 2.3.3, “Basic Configuration” on page 32 to see how to add DHCP options.

4.5.4.6 Integrating DNS and WINS

In 4.5.4.2, “Verifying TCP/IP Configuration” on page 102, we discussed how to set up a static DNS server. Assuming you now have the DHCP server and the static DNS server running in your network, what happens to name resolution when a host releases its IP address and obtains a different IP address next time? Since DNS is a static database, the administrator must update it manually. If you have DHCP server in your network and can assign IP configuration dynamically, don’t you expect the name server to update the database dynamically, too?

By integrating the static DNS server with the WINS server, you can resolve the names dynamically.

WINS (Windows Internet Name Service) servers maintain a database that maps NetBIOS computer names to TCP/IP addresses. WINS supports

dynamic computer name registration and name resolution. Each time a WINS client starts, it registers its NetBIOS name - IP address mappings with the configured WINS server. Although WINS provides dynamic name services, it offers a NetBIOS name space; so you need to integrate WINS server with the DNS server to resolve TCP/IP hostnames dynamically. This is achieved by having a DNS server to point to a WINS server for both forward and reverse name resolutions. By enabling WINS lookup, if a mapping is not found in the DNS database, DNS server queries a WINS database.

Figure 73 shows the interaction between a DHCP server, WINS server, and a static DNS server.

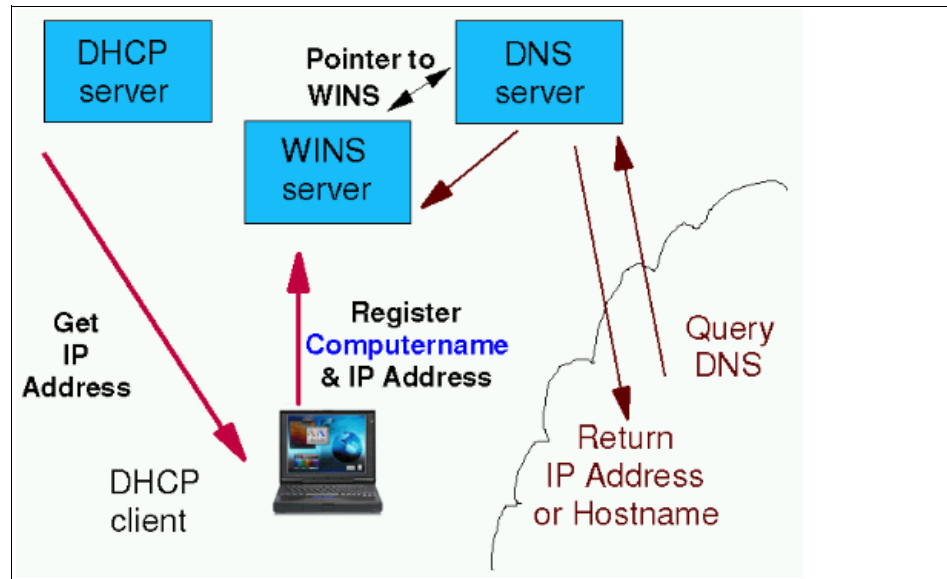


Figure 73. [Windows NT] DHCP – WINS – DNS Interaction

4.5.4.7 Installing WINS Server

If WINS server is not installed, follow the steps below:

1. Open the **Control Panel**.
2. Open the **Network** folder.
3. On the Network notebook click on the **Services** tab and click on the **Add...** button.
4. Select **Windows Internet Name Server** from the list and then click **OK**.

5. When prompted, enter the path to the Windows NT files (Windows NT 4.0 Server CD) and click on **Continue** to copy the files and finish the installation.

4.5.4.8 Configuring WINS Server

By default, the WINS server service starts automatically when the computer is started. If the WINS server is not running, open the **Service** icon in the Control Panel, select **Windows Internet Service** in the service list and click on the **Start** button. You can also start the WINS server at the command prompt by using the following command:

```
NET START WINS
```

To start the WINS Manager, select **Programs** from the Start menu, then select **Administrative Tools (Common)**, and then click **WINS Manager**. WINS Manager can show some basic statistics for the selected server. To display additional statistics, click **Detailed Information** on the Server menu. You can also view all the mappings by selecting **Show Database** from the Mappings menu.

To integrate the DNS server with a WINS server, you need to configure your DNS server to enable WINS lookup for both forward and reverse mappings. Follow the steps below:

1. Open the **DNS Manager**.
2. Highlight the zone (domain) for forward mappings. Select **Properties** from the DNS menu.
3. Click on the **WINS Lookup** tab. Check **Use WINS Resolution** and enter WINS server's IP address. Click **OK**.

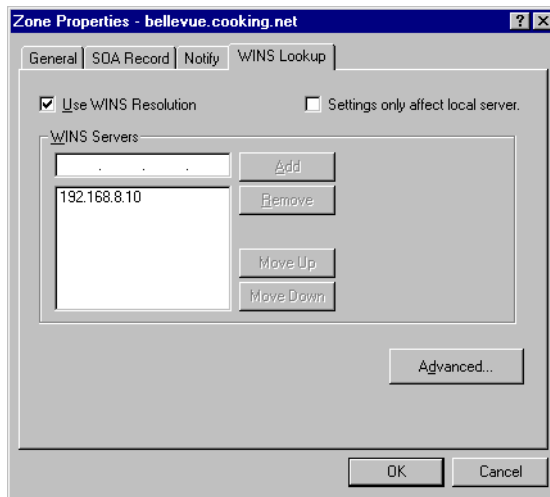


Figure 74. [Windows NT] Enabling WINS Lookup

4. Highlight the appropriate in-addr.arpa zone (domain) for reverse mappings. Select **Properties** from the DNS menu.
5. Click **WINS Reverse Lookup** tab. Check **Use WINS Reverse Lookup** and enter the DNS host domain to be appended to the NetBIOS name before returning the response. Click **OK**.

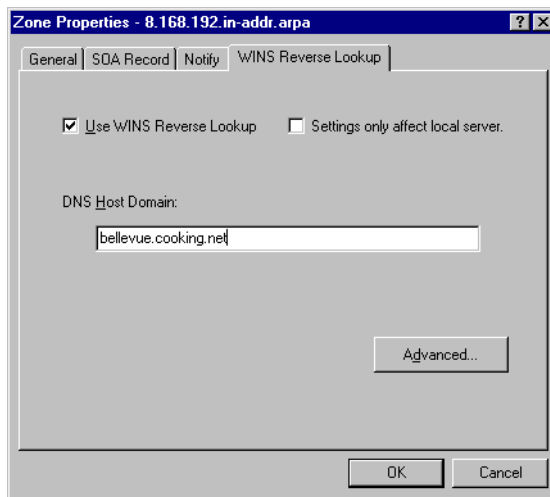


Figure 75. [Windows NT] Enabling WINS Reverse Lookup

6. You also need to add some DHCP options. To add the options, see 2.3.3, “Basic Configuration” on page 32. Add and configure the following options using the DHCP manager:

Option 044 WINS/NBNS Servers IP address of the WINS server

Option 046 WINS/NBT Node 0x8 (H-node)

Option 044 is used to configure the IP address of the WINS server. Option 046 is used to configure clients to use H-node. For more information about H-node, refer to 5.2, “Resolving NetBIOS Names to IP Addresses” on page 181.

7. Restart the DHCP, DDNS, and WINS servers.

4.5.4.9 Configuration Files

Table 5 shows DNS configuration files for the Windows NT server. After configuring a DNS server, these files are created and stored in the \WINNT\SYSTEM32\DNS directory.

Note: \WINNT is the default directory where Windows NT is installed.

Table 5. DNS Configuration Files

Name of the File	Description of the File
<zone_name>.dns	Database file Contains resource records for the zone Used for forward mappings
w.x.y.z.in-addr.arpa.dns	Reverse lookup file Maps IP addresses to hostnames
cache.dns	Contains names and addresses of the root domain name servers
Boot file	Used by manual start-up method *

* By default, the NT DNS server is configured to boot from the data in the registry.

Figure 76 on page 110 shows an example of a <zone_name>.dns file.

```

;
; Database file bellevue.cooking.net.dns for bellevue.cooking.net zone.
; Zone version: 41
;

@           IN      SOA      mustard.bellevue.cooking.net.
Administrator.bellevue.cooking.net.  (
    4           ; serial number
    3600        ; refresh
    600         ; retry
    86400       ; expire
    3600        )   ; minimum TTL

;
; Zone NS records
;

@           IN      NS      mustard

;
;WINS lookup record
;

@           0       IN      WINS      192.168.8.10

;
;Zone records
;

mustard           IN      A      192.168.8.10

```

Figure 76. [Windows NT] BELLEVUE.COOKING.NET.DNS File

Figure 77 on page 111 shows an example of a w.x.y.z.in-addr.arpa file.

```

;
; Database file 8.168.192.in-addr.arpa.dns for 8.168.192.in-addr.arpa zone.
; Zone version: 41
;

@           IN      SOA      mustard.bellevue.cooking.net.
Administrator.bellevue.cooking.net.(
        4          ; serial number
        3600       ; refresh
        600        ; retry
        86400      ; expire
        3600       ) ; minimum TTL

;
; Zone NS records
;

@           IN      NS       mustard.bellevue.cooking.net.

;
; NBSTAT lookup record
;

@           0        IN      WINS-R  bellevue.cooking.net.

;
; Zone records
;

10          IN      PTR      mustard.bellevue.cooking.net.

```

Figure 77. [Windows NT] 8.168.192.IN.ADDR.ARPA.DNS File

4.6 DNS Client Support in Windows NT

If you have computers with static IP addresses, you need to enable the DNS client function in the TCP/IP configuration panels.

4.6.1 Windows 95

To enable DNS client function in Windows 95, follow the steps below:

1. Open the **Control Panel**.
2. Double-click on the **Network** icon.
3. Highlight the **TCP/IP** protocol and click on the **Properties** button.
4. Click the **DNS Configuration** tab.
5. Check **Enable DNS**.
6. Enter a hostname and domain name. Enter IP address(es) of the DNS server(s) and click **Add**.

4.6.2 Windows NT Workstation 4.0

To enable the DNS client function for Windows NT Workstation 4.0, follow the steps below:

1. Open the **Control Panel**.
2. Double-click on the **Network** icon.
3. Click the **Protocols** tab, highlight **TCP/IP Protocol** in the Network Protocol list and click on the **Properties** button.
4. Click on the **DNS** tab. Enter a hostname and domain name. Click **Add**. In the DNS server box, enter IP address(es) of a DNS server(s) and click **Add**.

4.7 OS/2 Warp Server as Dynamic DNS Server

In Chapter 2, “Up and Running with DHCP” on page 17, we discussed how to set up and run a DHCP server. With a DHCP server in your network, IP addresses and other configuration information are assigned automatically to the clients. But one problem persists:

- How can a domain name server learn about those dynamically assigned IP addresses and hostnames so it can update its database accordingly?

One solution is to use a WINS server as discussed in 4.5.4.6, “Integrating DNS and WINS” on page 105. However, if you have clients other than Microsoft clients, they cannot register their names with WINS servers, and they cannot be reached by names. This problem can be solved by using the Dynamic Domain Name System (DDNS) services.

This chapter shows you how to set up your OS/2 Warp Server as a Dynamic DNS server. In our scenario, we use TCP/IP for OS/2 Version 4.1 which is the most current one at the time this book was written. Before we show you how to set up the server, we provide a brief introduction of DDNS in the following section.

4.7.1 The Dynamic Domain Name System (DDNS)

The DNS servers only support queries on a statically configured database, and the DNS administrators have to manually update this database whenever there is a change in the network and client configurations. The Dynamic DNS (DDNS) protocol defines an extension to this domain name system to update the DNS database dynamically. Using DDNS protocol, both the address (A) record, which maps a hostname to its IP address, and the pointer (PTR)

record, which maps a host's IP address to its hostname, can be added and deleted dynamically.

4.7.1.1 IBM's Dynamic Domain Name System

IBM's Dynamic Domain Name System (DDNS) is based on, and is a superset of, the Internet Software Consortium's publicly available implementation of the Berkley Internet Name Domain (BIND) level 4.9.3. DDNS supports both static and dynamic DNS domains. In dynamic domains, only authorized clients can update their own data. In IBM's DDNS, RSA public-key digital signature technology is used for client authentication.

With IBM's DHCP server, an administrator can configure host configuration parameters only at a server and can automate the configuration of IP hosts. DDNS provides dynamic hostname-to-IP address (and IP address-to-hostname) mapping for Dynamic IP clients. Using DDNS, clients automatically update their A record with their new address, obtained from the DHCP server, and IBM DHCP servers can automatically update PTR records for those clients.

4.7.1.2 DDNS Mechanism

With DDNS, both address (A) records and the pointer (PTR) records can be updated dynamically. The DDNS client program, NSUPDATE, is used to update information in a DDNS server.

Dynamic updates are performed by any of the following:

- **Network client:** A host which has DHCP and DDNS client software and can update its A and TXT records with the current IP address information.
- **DHCP server:** A host that updates PTR records with current hostname information for the address it allocates, and on certain occasions, updates A records for clients that either cannot or do not update the A records themselves.
- **DDNS system administrator:** A user who can update data, such as A and PTR records in a dynamic domain.

The interaction between a DDNS client (including DHCP server) and a DDNS server can be described as follows and is summarized in Figure 78 on page 114.

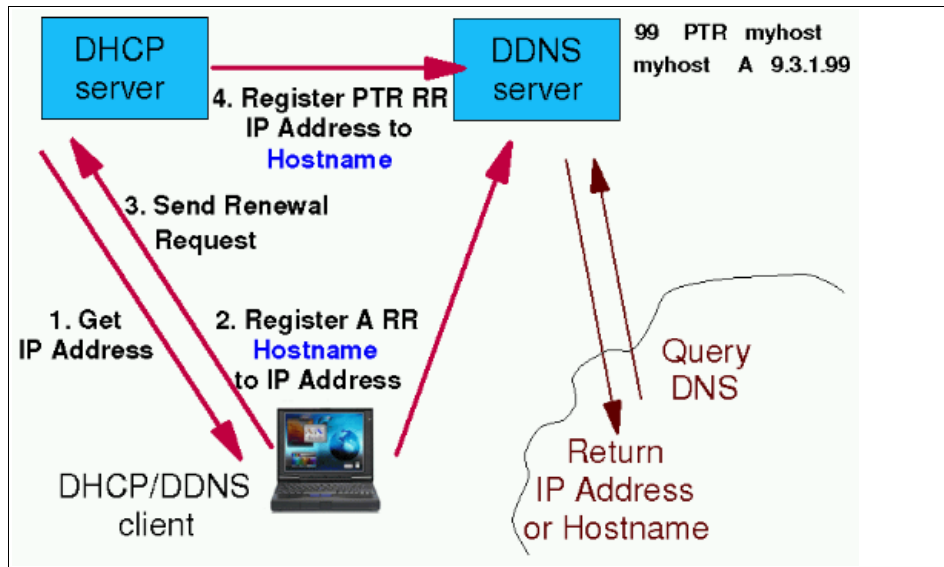


Figure 78. [Warp Server] DHCP – DDNS – Client Interaction

- After a DHCP/DDNS client has been provided with all the information by a DHCP server, it contacts a DDNS server.
- The DDNS client then sends an update request for the resource records which are associated with the client's hostname. The client also sends its public key and signs all resource records with a digital signature. The key and the signature, together, will allow the server to verify the authenticity of the update, as well as any other software which supports the MD5 algorithm. For more information, see 8.2, "RSA Public Key Authentication System" on page 310.
- If the client's updates have been successful, the server commits the changes to its database, and the client becomes reachable by its hostname.
- A name server normally supports inverse queries. This means the DDNS server must be updated with the reverse information. A DHCP server running on the OS/2 Warp server can update the reverse mappings for clients. After the DDNS client has successfully registered its name with the DDNS server, the DHCP client will send a lease renewal request message to the DHCP server. The client will include the new hostname in the message. The DHCP server is told to update the database with the client's new IP address and hostname.

4.7.1.3 Security

By having a DHCP and DDNS server in your network, hostnames can be registered automatically and updated dynamically. However, this means that without DDNS client authentication, an unauthorized host might impersonate an unsuspecting host by remapping the address entry for the unsuspecting host to that of its own and could intercept data (for example, logon passwords) intended for the unsuspecting host.

RSA digital signature technology is implemented for DDNS in TCP/IP for OS/2 to authenticate the owner of the DNS records and to secure the DNS database updates.

RSA Digital Signature Technology

RSA digital signature technology is based on the use of private-public key pairs:

- The private key is used to generate a unique digital signature that can be validated only by using the corresponding public key.
- The private key cannot be derived from either the digital signature or the public key or the combination of the two.

When a DDNS client registers its hostname for the first time, it generates an RSA key pair. A public key is sent to the DDNS server and registered for that particular hostname.

The DDNS client retains the RSA key pair (with the private key encrypted) in the client key file called DDNS.DAT (in the directory specified by the *ETC* environment variable). When the DDNS client updates the resource records on the DDNS server, the digital signature is generated by using the private key and is sent to the server with the update data.

When the DDNS server receives the update request, it uses the DDNS client's public key (in the KEY resource record) to:

- Authenticate the owner of the update request to verify that the update request was signed with the corresponding private key.
- Verify that the data was not changed since it was signed (In other words, to verify that no one intercepted and changed the data on the way to the name server.)

In this way, only the owners of the original records, who possess the correct private key necessary to generate the correct digital signature, can update resource records that are protected by an existing KEY resource record.

In addition, a DDNS administrator can create and use a zone key for each dynamic domain. The zone key is the administrator's RSA key pair for a particular domain. This key enables the administrator to create, modify, or delete any host's record in the domain, regardless of who created the records. The private key is used to generate a signature when the administrator updates the requests, and the server examines the signature to verify that update requests are from the administrator. The public key is registered in the domain data file as a KEY resource record for the domain. The private and public keys are stored in the DDNS.DAT file as the administrator's key file in the directory specified by the ETC environment variable.

For more information, see 8.2, "RSA Public Key Authentication System" on page 310.

Security Planning

As discussed in the previous section, DDNS servers use RSA digital signatures to authenticate DDNS updates requests. DDNS supports two modes of securing updates for a dynamic domain. Both modes protect the records in the database from unauthorized updates.

Dynamic Secured Mode This is a default mode. In the dynamic secured domain, DDNS clients can generate their own RSA key pair and dynamically register a KEY resource record containing the public key when they create their own name record for the first time. This means that servers allow any hosts with DDNS protocol to register their hostnames but once registered, only registered hosts (and administrators) are allowed to update their own entry.

Dynamic Presecured Mode This is an alternate mode of the secured mode. Figure 79 on page 117 summarizes dynamic presecured mode. In the presecured dynamic domain, DDNS clients must be pre-authorized by a DDNS administrator before they can create their name record. The DDNS administrator must preregister hosts and generate RSA keys for each client. This means that in presecured mode, the KEY resource record must be already defined in the domain before an update is accepted. The DDNS administrator must also distribute the correct corresponding key information to each

host before they create the resource records. See 8.4, "Presecured Domain" on page 315, for more information.

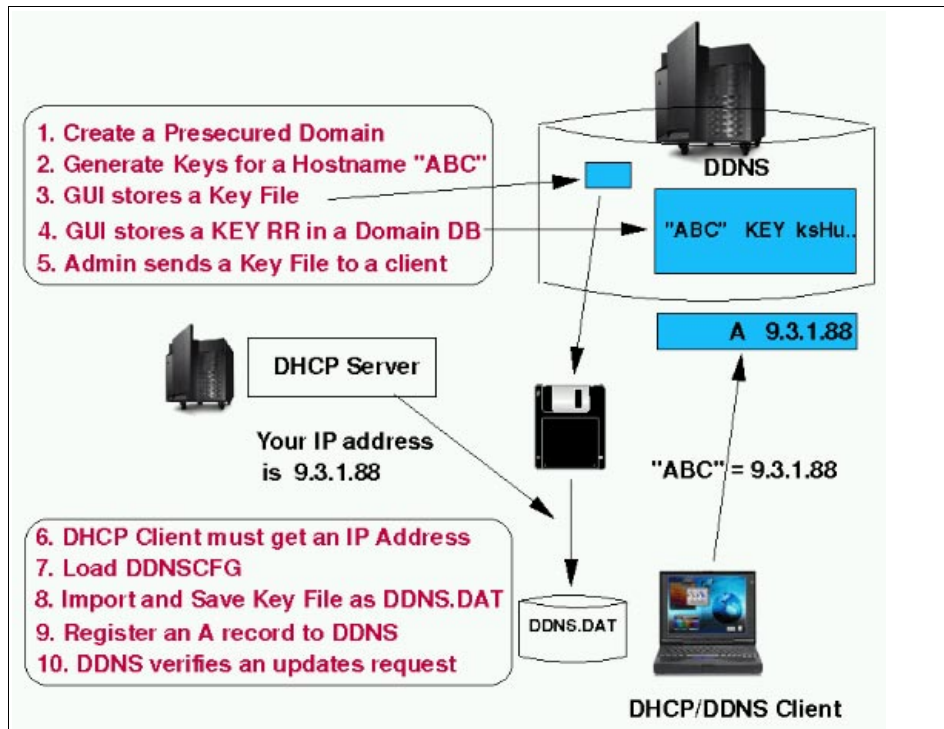


Figure 79. [Warp Server] Dynamic Presecured Domain

For clients that do not have DDNS client function, you can use a function called ProxyArec.

ProxyArec

For a client that can not update its A record by itself, a DHCP administrator can configure the DHCP server to update both the A and PTR records on behalf of the clients, as shown in Figure 80 on page 118. This function is called a ProxyArec. When using ProxyArec, a DHCP server uses a hostname sent by a client through Option 12 to update a DDNS server. For more information, refer to 8.5, "ProxyArec Consideration" on page 318.

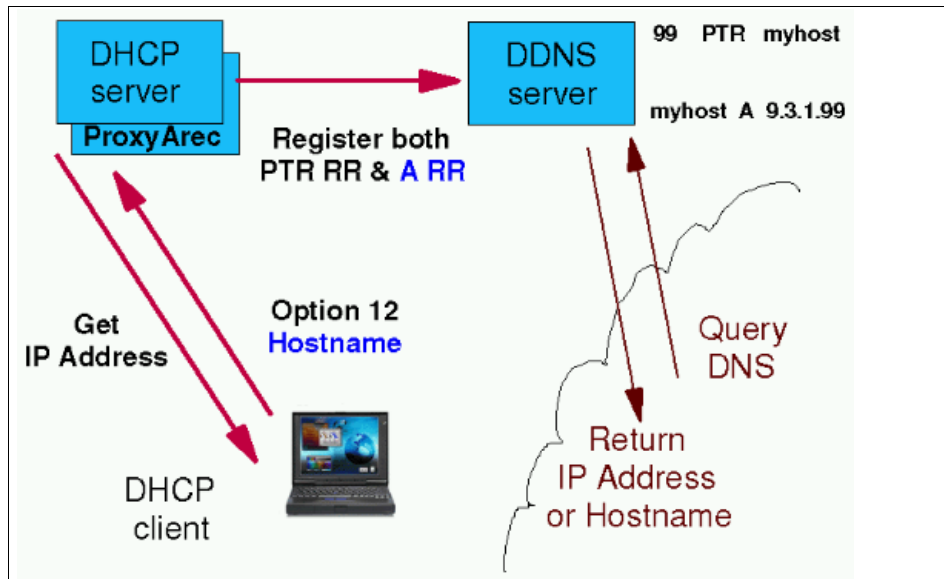


Figure 80. [Warp Server] Using ProxyArec

4.7.2 IBM's Dynamic DNS Relative to Other Implementations

IBM implemented Dynamic DNS in 1995. This implementation was done by IBM research based on an Internet draft being discussed in the IETF at that time. The protocol was going to become a standard in a very short time.

4.7.2.1 Basic Design

The basic design involves the following items:

- Allowing updates to the DNS by remote client machines.
- Having these updates be authenticated using RSA digital signatures, as illustrated in "RSA Digital Signature Technology" on page 115.
- Having the concept of expiration of resource records, so when an update was done in coordination with DHCP, the entries expired along with the IP address lease, thus making the system self administering.

There were also two RSA public-private key management mechanisms in the code:

- The secured mode, where the client could create/register its public key in the DNS with a hostname, and after that point, be the only authorized entity to update that information. You can find detailed information on this topic on page 116.

- The pre-Secured mode, where the systems administrator was the only authorized entity to create/register a public key with a hostname. After that time, the client machine that had the corresponding private key was the authorized entity to update that entry. There is more information on this topic on page 116.

The system was designed to integrate DHCP and DDNS, and more importantly, allow the client to manage its own A record in the DNS.

It is important to note that the IETF only standardizes protocol on the wire. The key management policies, the integration of DHCP and DDNS are all IBM design which differentiate an integrated product from a pure technology.

4.7.2.2 Current State of the Standards

The IETF did not standardize the draft which IBM research worked from. That standard evolved into the current RFC 2137 (Secure Dynamic Updates). Its resource records in the DNS are almost identical to IBM's implementation. However, the protocol on the wire is different. It also deals with the issue of integrating dynamic updates into secure DNS resolving (RFC 2065) and describes two modes of operation, mode A and mode B. In mode A, the client-signed RRs are stored as part of the DNS data, whereas in mode B, the server signs all the records. The IBM implementation of dynamic updates is actually very close to mode A (although the secure resolving described in RFC2065 is not supported yet).

The IETF also has an unsecure updates standard RFC 2136. (Currently in BIND 8.1.1).

There is also another Internet draft using Transaction Signatures (T-Sig) which uses shared secrets rather than public key cryptography to secure communication between the nameserver and a client. It seeks to be a simpler alternative to both RFC2065 for secure resolving and RFC2137 for dynamic updates, in those situations where the client and the server can establish a trust relationship. This is the mechanism Microsoft supposedly will use in Windows NT 5.0.

4.7.2.3 Differences in Different Standards

The following differences apply to IBM's implementation of secured Dynamic DNS versus RFC 2137, RFC2136, and T-Sig:

- **IBM DDNS and RFC 2137**

In normal cases, the logical evolution of IBM implementation would be to move towards RFC 2137 mechanism. This would not lead IBM to any new function, but IBM would have the comfort of stating that we comply with RFC 2137. This would become important if other vendors were developing clients or servers with this capability. No one has done that until now; so moving to this at this time does not get IBM any practical advantage over maintaining the current method.

- **IBM DDNS and RFC2136**

RFC 2136 was not designed for any authentication or expiration of records. It was also not designed for clients updating the DNS records. Its main design point was to allow, it does not have the following capabilities which were essential for our integrated product:

1. No real security mechanism. It accepts a list of IP addresses where the updates to any record in the system can originate (per zone). Thus, not only is this mechanism unsecure (for example, anyone who can update can update anyone's record), it is not scalable. One cannot realistically put every possible IP address all the clients may be sending, the requests from in this list. Some have tried to use this in a proxy-update mode where the DHCP server updates both the entries and the DHCP server's IP address is the only one allowed to update. This mechanism still has security flaws, since any machine can send any other machine's hostname as the requested name, and the DHCP server has no mechanism of verifying that information. The last one in, wins.
2. No mechanism of expiration. Thus, if a DHCP client got an address and then moved away from the network, the DHCP server's IP address lease would expire after a period of time and the address would get reused, but the DNS entry would have to be manually removed.
3. No mechanism to detect out-of-order update requests or prevent replay-attacks. (No timestamp associated with the update request.)

- **IBM DDNS and T-Sig**

T-Sig accomplishes the goal of providing an authentication mechanism for updates from clients. But it still does not provide an expiration mechanism. This means that if a client created its A-RR, and moved away from the network for some period of time. The PTR record created by the DHCP server could be deleted by it, but the A-RR (the hostname to IP address mapping) would remain in the DNS. So if someone else connected to the network and got assigned the same IP address by the DHCP server (which is legal since the lease expired), that machine also got a hostname which it did not have access to.

T-Sig also has scalability issues, since the shared secret keys are theoretically unique between every client-name server pair and must be predefined in each.

T-Sig does have a timestamp associated with its update requests, but since these timestamps are not stored with the data updated, it is questionable how out-of-order update requests or replay attacks could be effectively detected and/or prevented.

4.7.2.4 Conclusion

Thus, moving to the BIND 8.1.1 or the T-Sig capability actually introduces flaws in the update system and are not recommended from a system design perspective.

4.7.3 Scenario

In this section, we use an example of a simple network environment which only includes one subnet. In this network, we have one OS/2 Warp Server 4.0 with a DHCP server (See 2.2, “OS/2 Warp Server as a DHCP Server” on page 18 for how to set up a DHCP server) running and another OS/2 Warp Server with WorkSpace On-Demand manager running. We also have a few DHCP/DDNS client computers and WSOD clients. When we started building our network, we set up a DHCP server and initially did not have a DNS/DDNS server. In this section we will integrate DDNS into our network.

In this scenario, we configure a OS/2 Warp Server as a Dynamic DNS server. We configure a DDNS server in two cases:

1. Dynamic Secured Mode
2. Dynamic Presecured Mode

Later in this chapter, we also configure our DHCP server to enable ProxyArec and have Windows 95 and WSOD clients configured through this function.

In our example, our domain name is armonk.cooking.net, and the DHCP/DDNS server's hostname is merlot. Table 6 shows our network configuration:

Table 6. OS/2 Warp Server Configuration Information

Configuration Settings	Value
Network ID	192.168.6.0
Subnetmask	255.255.255.0

Configuration Settings	Value
Domain Name	armonk.cooking.net
DHCP/DDNS Server IP address	192.168.6.10
DHCP/DNS/WINS Server Hostname	merlot

4.7.4 Tasks

The following list describes the tasks to complete the scenario, followed by detailed, step-by-step instruction sections.

1. Plan the primary domain.
2. Define your OS/2 Warp Server as a primary DNS server for your DDNS domain.
3. Create a primary domain for forward mappings.
4. If using presecured mode, generate keys for the clients.
5. If there are hosts using a static IP address, add static records for them.
6. Configure a DHCP server to update PTR records.
7. Start the DDNS server
8. Restart the DHCP server.
9. Verify that the DDNS server is operating correctly.

4.7.5 Planning

Before you start configuring your Dynamic DNS server, you need to make a plan on the design of your network that includes domain and security issues.

4.7.5.1 Designing your Domain and Domain Name Server

The first step is to design your domain. Before you establish DNS servers in your network, you need to determine the domain name for your domain. Furthermore, you need to determine the server that is going to be the Dynamic DNS server for your domain.

In our scenario, armonk.cooking.net is the primary domain name, and Merlot is the name of the name server. All hosts on the 19.168.6.0 network are included in the domain armonk.cooking.net.

Then, you need to decide which hosts are going to be the DHCP/DDNS clients and which are not. A DDNS name server must have a static IP address and, therefore, cannot be a DHCP client. If you have any other hosts

that have static IP configuration, you have to decide on their names and configure the DNS database manually.

4.7.5.2 Security Planning

The next step is to decide on security issues. Dynamic DNS in TCP/IP Version 4.1 for OS/2 supports two modes of security:

- Dynamic Secured Mode
- Dynamic Presecured Mode

To decide which mode to implement, refer to 4.7.1.3, “Security” on page 115 and also see Chapter 8, “Security of DHCP and Dynamic DNS” on page 309. If you have clients that do not support DDNS, you need to enable the ProxyArec function at the DHCP server. For more information about ProxyArec, refer to 8.5, “ProxyArec Consideration” on page 318.

4.7.6 Setting Up your DDNS Server

This section describes how to set up a DDNS server on your Warp server computer. 4.7.6.2, “Dynamic Secured Mode” on page 124 shows how to set up DDNS server with dynamic secured mode and 4.7.6.3, “Dynamic Presecured Mode” on page 130 shows you how to set up with Dynamic Presecured mode. All values we used to configure in the following steps are the values we defined in our scenario.

4.7.6.1 Things To Do Before Setting Up

Before you start configuring, make sure that the TCP/IP is configured correctly. The Dynamic DNS server must have a static IP address, and therefore, cannot be a DHCP client.

The Dynamic DNS server computer itself can also be a DNS client. Follow the steps below to add the DNS server information:

1. Open the **TCP/IP Shadows** folder and the **TCP/IP Configurations** folder and then double-click on **TCP/IP Configuration (Local)** or type `TCPCFG2` at an OS/2 command prompt. The TCP/IP configuration settings notebook is displayed.
2. Click on the **Host Names** tab. Type hostname and local domain name.
3. Click **Add**. Type an IP address for the computer (DNS/DDNS server) and click on **OK**.
4. Close and save the configuration.

Important

The next step is very important. You need to create an entry in the HOSTS file. DDNS Administration GUI tool uses information in the HOSTS file, and this is required to automatically create a primary zone for reverse

1. Open the HOSTS file using an ASCII editor. The HOSTS file is stored in the directory specified by the `ETC` environment variable. (If not certain, type `SET ETC` to check which directory the `ETC` environment variable points to.)
2. Add your name server hostname using the fully-qualified domain name. Figure 81 shows you an example of the HOSTS file after adding a new entry.

127.0.0.1	localhost
192.168.6.10	merlot.armonk.cooking.net

Figure 81. [Warp Server] HOSTS File

4.7.6.2 Dynamic Secured Mode

Follow the steps below to set up the DDNS server using dynamic secured mode:

1. You will use the DDNS Server Administrator program to configure the DDNS server. To do so, open the **TCP/IP Shadows** folder, then open **DDNS Server Services** and double-click on the **DDNS Server Administrator** icon.
2. DDNS Server Administrator window is displayed. Enter the TCP/IP administrator password and click **OK**.
Two settings notebooks are displayed.
3. At the Domain Name Server settings notebook, click the **Server** tab. Verify the name and IP address of the name server. (The GUI retrieves the hostname and IP address from the HOSTS file.)

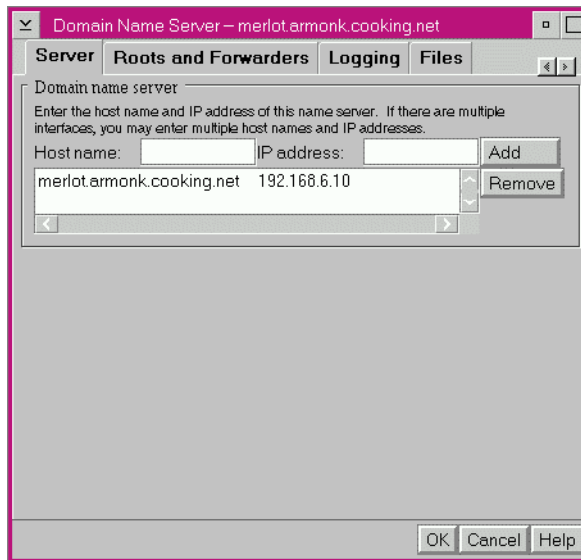


Figure 82. [Warp Server] Domain Name Server Notebook

If desired, on the Files tab, specify a different cache file name. Click **OK** to close the settings notebook.

4. At the DDNS Server Administrator window, define the DDNS server as the primary server for your domain as follows:
 - Click **Add Primary Domain** on the tool bar to open the Primary Domain notebook.
 - At the Domain Configuration tab, type your complete domain name.

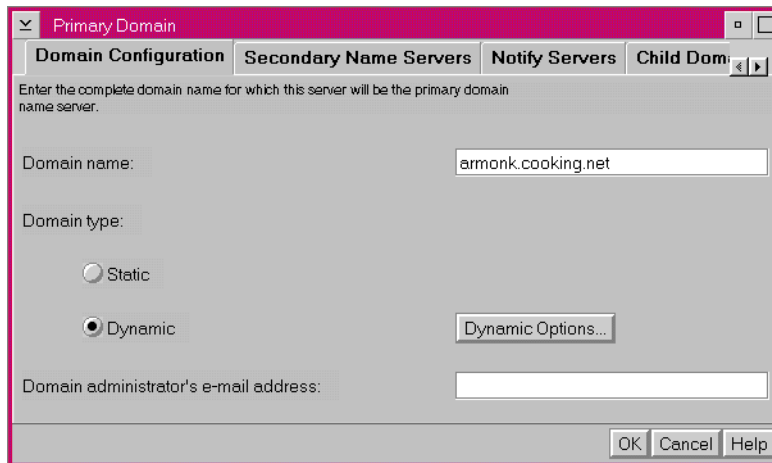


Figure 83. [Warp Server] Primary Domain Configuration

- At the Domain Configuration tab, note that the domain type is dynamic by default, as shown in Figure 83. Click on **Dynamic Options** and note that **Client automatically creates its own hostname** (which means Dynamic Secured mode) is checked, as shown in Figure 84 on page 126. Click **OK** to go back to the Domain Configuration tab.

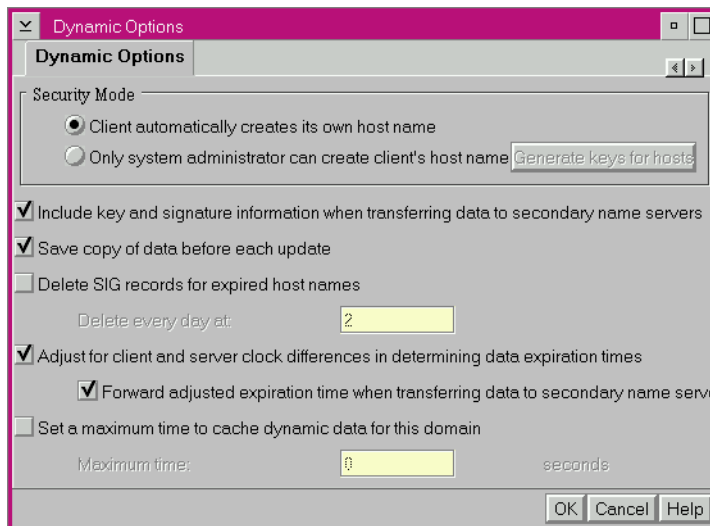


Figure 84. [Warp Server] Setting Secured Mode

- If you do not want to create a reverse mappings file (by default), click the **Domain Options** tab and uncheck **Automatically Create Reverse Mappings for Statically Defined Hosts**.
 - If you want to change the name of the file for your domain, click on the **Files** tab and type the path and the file name of the domain data file.
 - After you finish the configuration, click **OK** to close the notebook. When you close the Primary Domain Notebook, a zone key is automatically created.
5. When you create a primary domain, an A record, PTR record, and an alias ns-updates for the name server is created automatically, as shown in Figure 85 on page 127. If you do not want to use this alias, delete the alias or use the Alias Notebook to add a different alias.

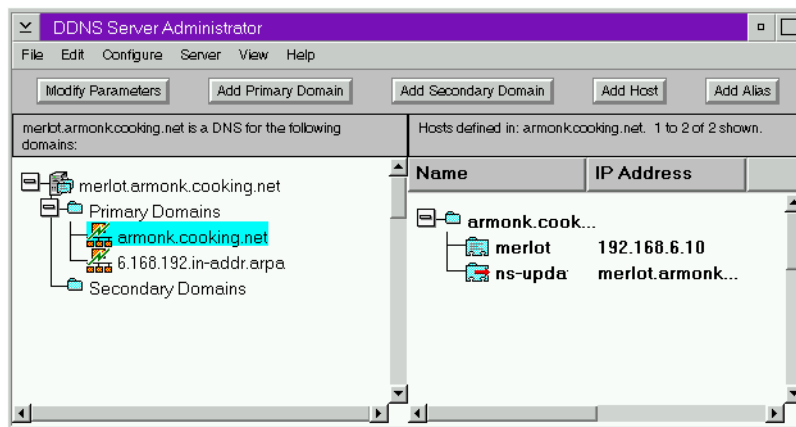


Figure 85. [Warp Server] DDNS Server Administrator Notebook

ns-updates Alias

A DDNS client must know the fully qualified domain name or IP address of the primary DDNS server. DDNS clients using the DDNS client configuration program for a DHCP client (DDNSCFG.EXE) defaults the name of the Primary DDNS server to ns-updates.<domain_name> (In our scenario, ns-updates.armonk.cooking.net.) An end user can change this ns-updates default at any time by using the DDNS client configuration program. If you follow the steps to modify HOSTS file, as described in 4.7.6.1, “Things To Do Before Setting Up” on page 123, this alias ns-updates is created automatically.

6. If you want to add any static hosts, follow the steps below:

- On the DDNS Server Administrator notebook, highlight the domain in which you are going to add a new host, and click **Add Host** on the tool bar.
- Type the hostname and IP address of the computer and click **Add**. Click **OK** when finished.

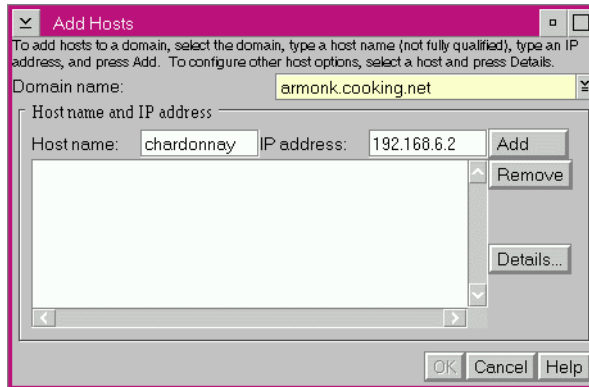


Figure 86. [Warp Server] Adding Hosts

7. Click **Save** from the File menu to save the configuration.
8. You need to configure the DHCP server so that DHCP clients can get the configuration information for DDNS/DNS server and domain name. Add following two options:
 - **Option 6 Domain Name Server:** IP address of the DNS/DDNS server
 - **Option 15 Domain name:** Domain name

Refer to 2.2, “OS/2 Warp Server as a DHCP Server” on page 18 to see how to add DHCP options.
9. If you want to update PTR records for DHCP/DDNS clients automatically, configure your DHCP server, as described in the steps below:
 - Open **DHCP Server Service** in the TCP/IP Shadows folder and double-click on **DHCP Server Configuration**.
 - Enter your password and click **OK**. The DHCP Server Configuration window comes up.
 - In the Current Configuration lists, double-click on your subnet to open the subnet notebook.
 - Click on the **Miscellaneous** tab, type the IP address of the DDNS server in the **DDNS Server for PTR** field.

The screenshot shows the 'Subnet' configuration window with the 'Miscellaneous' tab selected. The window contains several sections with configuration options:

- Automatically update DDNS A record on client's behalf if requested?**
 - Use default inherited from Global: No
 - Verify client ID before performing update? Yes
- If client omits a domain name, use server-assigned domain name?**
 - Use default inherited from Global: Yes
- Transform 802.3 MAC addresses to canonical form?**
 - Use default inherited from Global: No
- Next bootstrap server**
 - ☒ Use default inherited from {no inherited value}:
 - ☐ Specify value for this item: (empty text field)
 - IP address or hostname
- DDNS server for PTR record updates**
 - 192.168.6.10
 - IP address or hostname

At the bottom right, there are buttons for 'OK', 'Cancel', and 'Help'.

Figure 87. [Warp Server] Specifying DDNS Server for PTR Record Update

- Click **OK** to close the notebook.
- Repeat steps above for all the subnets you want to update PTR records dynamically.
- In the Current Configuration lists, double-click on **DHCP Server** to open the DHCP Server Parameters notebook.
- Click on the **DDNS PTR Records** tab, check **Automatically update or delete PTR recorded updates**, and click **OK**.

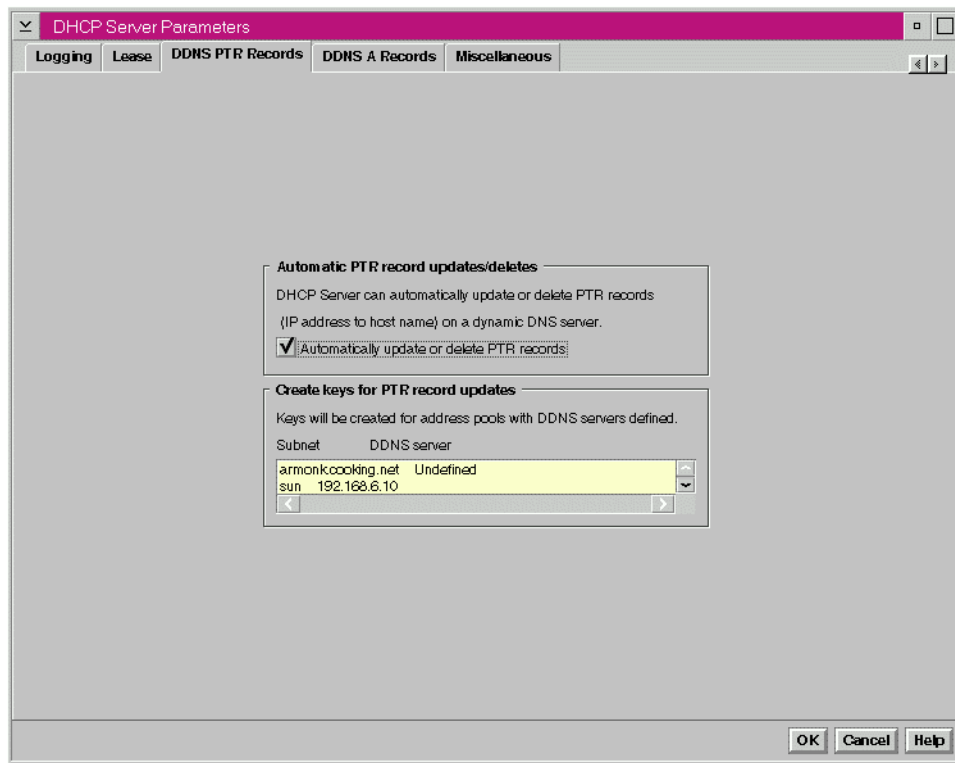


Figure 88. [Warp Server] Enabling DHCP Server to Update PTR Records

- Select **[File —Save]** to save the configuration in the DHCP Server Configuration window.
10. Double-click the **DDNS Server** in the DDNS Server Service folder to start the DDNS server.
 11. Restart the DHCP server.

4.7.6.3 Dynamic Presecured Mode

In this section, we show you how to setup a dynamic presecured domain using the DDNS Server Administrator program.

1. Open the **TCP/IP Shadows** folder, then **DDNS Server Services**, and double-click on the **DDNS Server Administrator** icon.
2. The DDNS Server Administrator window comes up. Type the password and click **OK**.
3. Two notebooks come up. On the Domain Name Server notebook (Figure 89 on page 131), click the **Server** tab. Verify the name and IP address of

the name server. (GUI uses the hostname and IP address from the HOSTS file.)

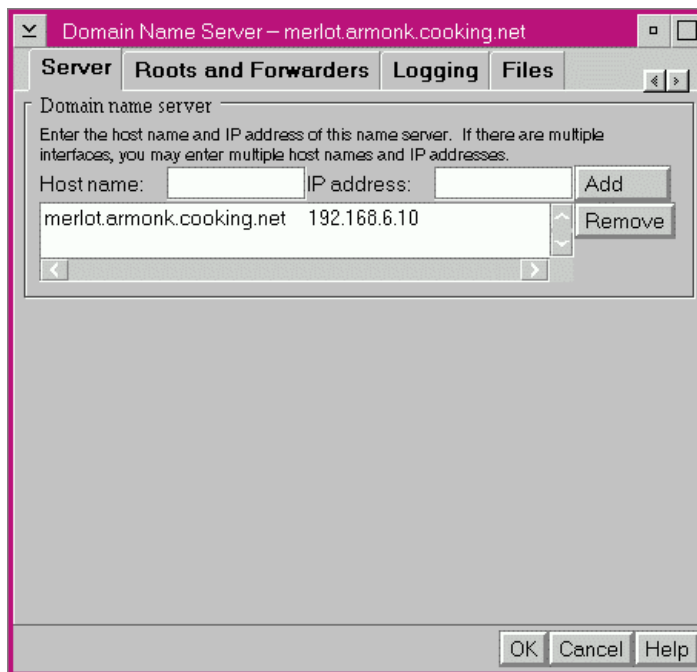


Figure 89. [Warp Server] Domain Name Server Notebook

If desired, on the Files tab, specify a different cache file name. Click **OK** to close the notebook.

4. Using the DDNS Server Administrator window, define the DDNS server as a primary server for your domain as follows:
 - Click **Add Primary Domain** on the tool bar to open the Primary Domain notebook.
 - On the Domain Configuration tab, type your complete domain name.

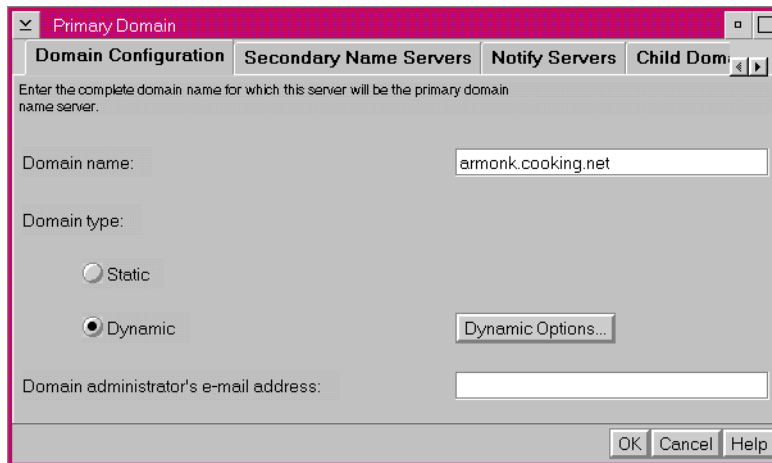


Figure 90. [Warp Server] Primary Domain Configuration Notebook

- On the Domain Configuration tab, note that the Domain Type is set to be Dynamic, as shown in Figure 90. To set the mode to be presecured, click **Dynamic Options** to go to the Dynamic Options Notebook and then check **Only System Administrator Can Create Client's Host Name** (dynamic presecured mode) as shown in Figure 91.

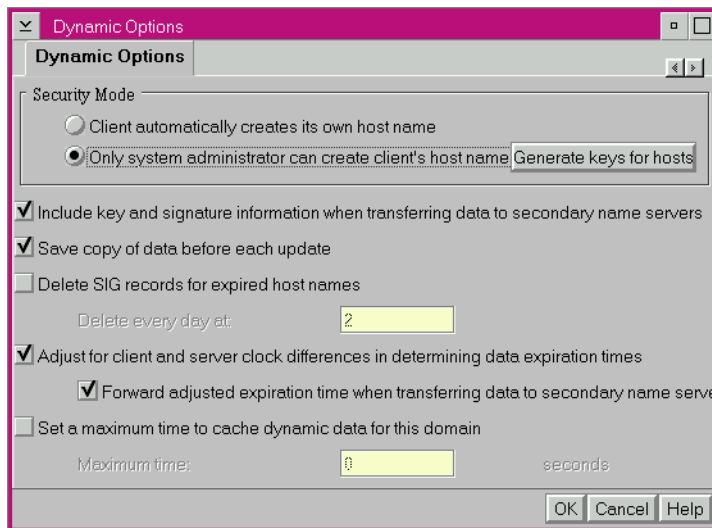


Figure 91. [Warp Server] Setting Dynamic Presecured Mode

- If you do not want to create a reverse mapping (by default), click the **Domain Options** tab and uncheck the **Automatically Create Reverse Mappings for Statically Defined Hosts**.
 - If you want to change the name of the file for your domain, click on **Files** tab and type the path and file name of the domain data file.
 - After you finished your configuration, click **OK** to close the notebook. When you close the notebook, the zone key for the domain is created automatically.
5. When you create a primary domain, an A record, PTR record, and an alias ns-updates for the name server is created automatically, as shown in Figure 92. If you do not want to use this alias, delete the alias or use the Alias Notebook to add a different alias.

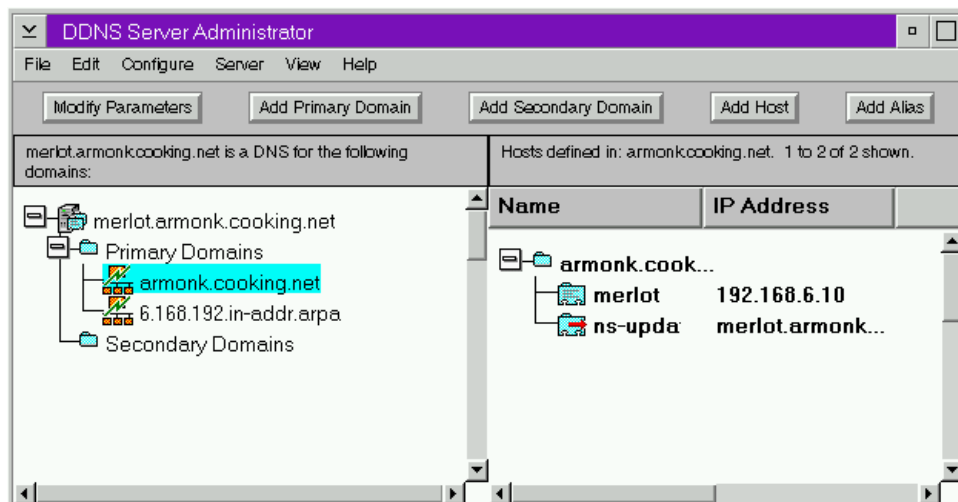


Figure 92. [Warp Server] DDNS Server Administrator Notebook

ns-updates Alias

A DDNS client must know the fully qualified domain name or IP address of the primary DDNS server. DDNS clients using the DDNS client configuration program for a DHCP client (DDNSCFG.EXE) defaults the name of the primary DDNS server to ns-updates.<domain_name> (In our scenario, ns-updates.armonk.cooking.net). An end user can change this ns-updates default at any time by using the DDNS client configuration program. If you follow the steps to modify HOSTS file, as described in 4.7.6.1, “Things To Do Before Setting Up” on page 123, this alias ns-updates is created automatically.

6. If you want to add any static hosts, follow the steps below:
 - In the DDNS Server Administrator window, highlight the domain in which you want to add a new host and click **Add Host** on the tool bar.
 - Type the hostname and the IP address of the computer and click **Add**. Click **OK** to finish the window.

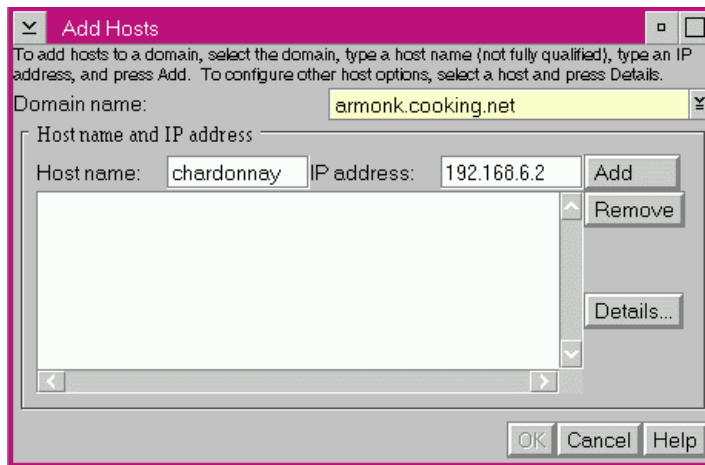


Figure 93. [Warp Server] Adding Hosts

7. Click **Save** from the File pull-down menu to save all configuration information.
8. Double-click on the **DDNS Server** object in the DDNS Server Service folder to start the DDNS server.
9. The last step is to register DDNS clients and generate host keys for them (the DDNS server must be started to generate keys):
 - Start the DDNS Server Administrator program.
 - Double click the domain for forward mappings in the list to open the Primary Domain notebook.
 - On the Domain Configuration tab, click on the **Dynamic Options** button.
 - Click **Generate Keys for Hosts** on the Dynamic Options Notebook.
 - On the Generate Keys Window, type hostnames and click **Add** for each client. If you want to change the directory where keys are stored, specify the path. Click **OK** to return to the Dynamic Options Notebook.

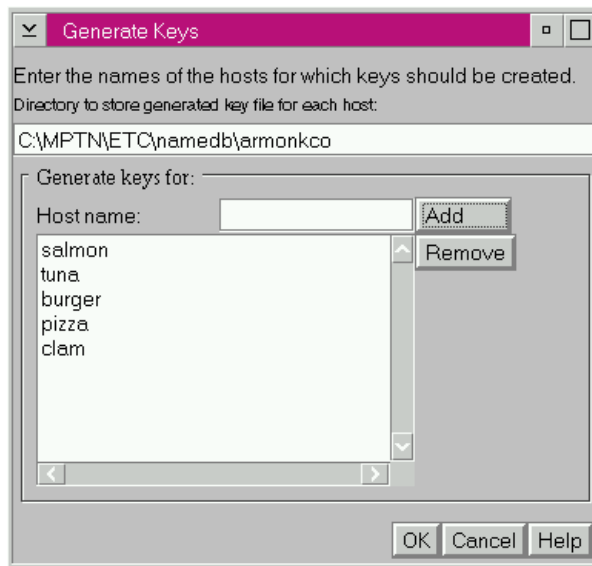


Figure 94. [Warp Server] Generate Keys for DDNS Clients

10. Click **OK** to close the Dynamic Options settings notebook.
11. Click **OK** to close the Primary Domain settings notebook. When you close the Primary Domain settings notebook, the host keys that you registered are created automatically under the directory you specified.
12. You need to configure the DHCP server so that DHCP clients can get the configuration information for the DDNS/DNS server and domain name. Add following two options:
 - **Option 6 Domain Name Server:** IP address of the DNS/DDNS server
 - **Option 15 Domain name:** Domain name

Refer to 2.2, “OS/2 Warp Server as a DHCP Server” on page 18 to see how to add DHCP options.
13. If you want to update PTR records for DHCP/DDNS clients automatically, configure a DHCP server as described in the steps below:
 - Open **DHCP Server Service** in the TCP/IP Shadows folder and double-click on **DHCP Server Configuration**.
 - Enter your password and click **OK**. The DHCP Server Configuration window comes up.
 - In the Current Configuration lists, double-click on the **subnet** <subnet name> to open the subnet notebook.

- Click on the **Miscellaneous** tab and type an IP address of the DDNS server in the DDNS Server for PTR Record update field.

The screenshot shows the 'Subnet' configuration window with the 'Miscellaneous' tab selected. The window contains the following settings:

- Automatically update DDNS A record on client's behalf if requested?**: Set to 'No' (dropdown). Below it, 'Verify client ID before performing update?' is checked.
- If client omits a domain name, use server-assigned domain name?**: Set to 'Yes' (dropdown).
- Transform 802.3 MAC addresses to canonical form?**: Set to 'No' (dropdown).
- Next bootstrap server**: The radio button 'Use default inherited from {no inherited value}:' is selected.
- DDNS server for PTR record updates**: The text field contains '192.168.6.10'.

Buttons at the bottom: OK, Cancel, Help.

Figure 95. [Warp Server] Specifying DDNS Server for PTR Record Update

- Click **OK** to close the settings notebook.
- Repeat the above steps for all the subnets you want to update PTR records dynamically.
- In the Current Configuration lists, double-click on **DHCP Server** to open the DHCP Server Parameters settings notebook.
- Click on the **DDNS PTR Records** tab, check **Automatically update or delete PTR records**, and click **OK**.

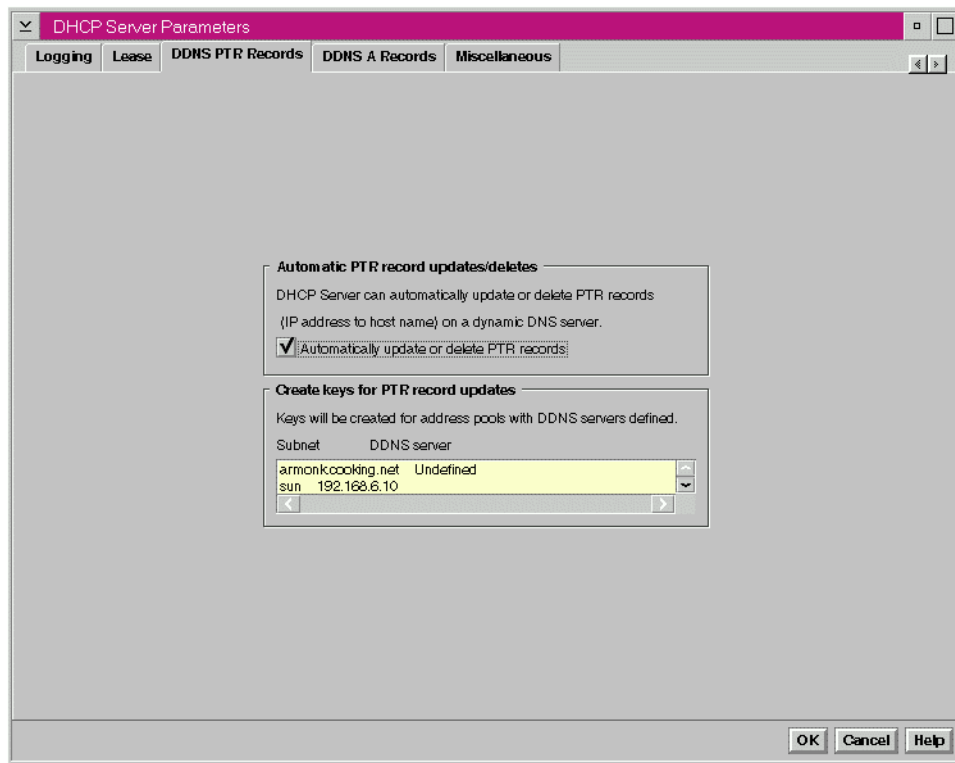


Figure 96. [Warp Server] Enabling DHCP Server to Update PTR Records

- Select [**File — Save**] to save the configuration in the DHCP Server Configuration window.
- Restart the DDNS server.
 - Restart the DHCP server.

4.7.7 ProxyArec

If you have any clients that do not have the DDNS client function, you need to configure your DHCP server to update A records on behalf of the DHCP clients.

To enable ProxyArec on your DHCP server, see 8.5, “ProxyArec Consideration” on page 318.

4.7.8 Verification

The last step is to make sure that your DDNS server is working properly. To verify the configuration, you can do the following:

- Verify that the name server can resolve names.
- Check the NSUPDATE log file.

4.7.8.1 Verifying the DDNS Server Configuration.

Use the `HOST` and `NSLOOKUP` commands to check whether the DDNS server can resolve names.

The `HOST` command contacts a DNS server to translate a specified hostname to its IP address or specified IP address to its hostname. For syntax information on the `HOST` command, refer to 10.2.6, “The HOST Command” on page 348.

`NSLOOKUP` queries a DNS server in either interactive or noninteractive mode. `NSLOOKUP` command syntax information is provided in 10.2.7, “The NSLOOKUP Command” on page 348.

4.7.8.2 Checking the NSUPDATE Log

The `NSUPDATE` agent, when used for a DHCP client, or server, logs all significant events to the `NSUPDATE.LOG` file. This file is stored in the directory specified by the `ETC` environment variable.

4.7.9 DDNS Files

This section describes the configuration files used by the DDNS server. The configuration files used by the DDNS server are:

- Configuration file
- Boot file
- Cache file
- Domain data files

These files are created and updated automatically when you configure your domain using the DDNS Server Administrator program. It is possible to create these files manually, but we recommend you use the DDNS Server Administrator program, because this program checks the configuration to see if there are any mistakes in the configuration you defined. It also automatically updates the serial number so that the secondary name server can notice that there has been a change in the database. In addition, by using

GUI, you do not need to execute the `DDNSZONE` command to generate keys, because the program automatically does it for you.

4.7.9.1 Configuration File DNSEXT.CFG

Many of the new enhancements IBM has added to the standard BIND DNS code require configuration. These new features are configured in a separate file, `\MPTN\ETC\NAMEDB\DNSEXT.CFG`, so that there is no confusion with standard name server configuration files.

The DDNS server GUI creates the `DNSEXT.CFG` file. It has a few entries that it places in there for its own information. However, this file is completely optional, since all parameters have default values.

Figure 97 on page 140 shows an example of the `DNSEXT.CFG` file.

```

; ***** IBM DDNS Server Administrator *****
; This file was written by the IBM DDNS Server Administrator on 28-Jun-98
; ***** IBM DDNS Server Administrator *****
austin.cooking.net (
  notify=yes
  notify.add=192.168.6.2
  ;notify.add=192.168.7.2 (This line is commented)
  notify.delayTime=60
  notify.retryTime=30
  notify.retryNumber=3
  timeSync=yes
  timeSync.toSecondaries=yes
  safeWrite=yes
  sigDel=no
  ttlSet=no
  deferUpdCnt=100
  incrTime=300
  keyToSec=yes
  sepDynStatic=yes
  reverseMapping=yes ;Used by the DDNS GUI only)
)
6.168.192.in-addr.arpa (
  notify=yes
  notify.delayTime=60
  notify.retryTime=30
  notify.retryNumber=3
  timeSync=yes
  timeSync.toSecondaries=yes
  safeWrite=yes
  sigDel=no
  ttlSet=no
  deferUpdCnt=100
  incrTime=300
  keyToSec=yes
  sepDynStatic=yes
  reverseMapping=yes
)
DDNSAdministratorClient (
  gui.warn=yes
  gui.write=yes
  gui.num=100
  gui.lease=3600
  gui.pad=3110400
  gui.reinit=1
  gui.sepdata=3
)

```

Figure 97. [Warp Server] DNSEXT.CFG

where: (see Table 7 on page 141).

Table 7. DNSEXT.CFG Configuration Parameters

Configuration Parameters	Parameter	Values	Description
Notify Parameters	notify	yes no	All notify parameters apply to both secondary and primary zones. This parameter must be set to yes in order for the other notify parameters listed below to apply. The default is no..
	notify.add	IP Address	Optional parameter which can be used to add a particular DDNS server's IP address to the list of servers to notify.
	notify.remove	IP Address	Optional parameter which can be used to remove a particular DDNS server's IP address from the list of servers to notify.
	notify.delayTime	number of secs	This is the amount of time between notifications sent to consecutive servers on the notify list. The actual delay time is a random time between the specified value and twice that value. The default value is 30.
	notify.retryTime	number of secs	This is the amount of time between notifications sent to a particular server. (Multiple notifies will be sent to a server if that server does not respond.) The default value is 60.
	notify.retryNumber	number	This is the maximum number of times that a notification will be retried to a particular server. The default value is 3.

Configuration Parameters	Parameter	Values	Description
Client/Server Time Synchronization Parameters	timeSync	yes no	This parameter defines whether the time synchronization feature is enabled or not, and it applies only to primary dynamic zones. The default value is yes.
	timeSync.toSecondaries	yes no	This parameter defines whether the time synchronization being kept should be forwarded to the secondary servers or not. This parameter will be ignored if the timeSync is set to no. The default value is yes.
SIG RR Deletion Parameters	sigDEL	yes no	This parameter defines whether the signature deletion function, which deletes signature records for expired host names, should be enabled (yes) or not (no), and applies only to the primary dynamic zones. The default value is no.
	sigDEL.time	number, (0 -23) military time.	This parameter defines the military time at which any signature records still remaining for the zone will be deleted. This parameter is ignored if sigDEL is set to no.
Safe Write to Disk for Zone Data Parameter	safeWrite	yes no	This parameter defines whether the safe write feature, which keeps a backup of the zone data, should be enabled (yes) or not (no). The default value is yes.

Configuration Parameters	Parameter	Values	Description
Override TTL for Dynamic Data Parameters	ttlSet	yes no	This parameter defines whether the TTL override function, which allows administrators to lower the TTL value for dynamic clients, should be enabled (yes) or not (no). This parameter applies only to primary dynamic zones. The default value is no.
	ttlSet.value	number	The number is the actual TTL value to use in the override of dynamic client TTLs. This parameter will be ignored if ttlSet is set to no. It is required if ttlSet is set to yes. The valid numbers for a TTL are the positive values of a signed 32-bit number.
Separate Static / Dynamic Zone Data Parameters	sepDynStatic	yes no	This parameter defines whether the static/dynamic data separation feature, where static and dynamic data for a dynamic zone are kept separately, is enabled (yes) or not (no). This parameter applies only to primary dynamic zones.

Configuration Parameters	Parameter	Values	Description
Configuration Parameters for Interoperability with earlier versions of DDNS	keyToSec	yes no	If keyToSec is set to no, it corresponds to nokeytosec on the primary line of the boot file in earlier versions of DDNS.
	incrTime	number of seconds	The default value is 300 seconds (5 minutes). This value corresponds to the optional 6th parameter of the SOA RR for primary dynamic zones from earlier releases of DDNS. It specifies the maximum amount of time to wait after an update before incrementing the zone serial number.
	deferUpdCnt	number	The default value is 100. This value corresponds to the optional 7th parameter of the SOA RR for primary dynamic zones from earlier releases of DDNS. It specifies the maximum number of updates which can occur before the zone serial number is incremented.
Parameters used by the DDNS Server GUI.	reverseMapping	yes no	This attribute is placed inside normal domain entries.
	DDNSAdministratorClient		This value is set up in the form of a domain which is used by the GUI. This domain and all its entries are created and used exclusively by and for the GUI.

4.7.9.2 Boot File

This file is read by the name server when it is started. The default name of this file is NAMED.BT or NAMED.BOOT and stored in the \MPTN\ETC\NAMEDB directory. This file includes information, such as the domain names for which this name server is authoritative and the location of the domain data.

Figure 98 shows an example of the NAMED.BT file.

```
;***** IBM DDNS Server Administrator *****  
; This file was written by the IBM DDNS Server Administrator on 28-May-98  
;***** IBM DDNS Server Administrator *****  
primary 6.168.192.in-addr.arpa C:\MPTN\ETC\namedb\dnsf0000.rev dynamic secured  
primary armonk.cooking.net C:\MPTN\ETC\namedb\dnsf0000.dom dynamic secured  
cache . C:\MPTN\ETC\namedb\named.ca
```

Figure 98. [Warp Server] NAMED.BT File

4.7.9.3 Cache File

The cache (root server) file contains the names and IP addresses of the root servers. The path and name of this file are specified in the cache entry in the boot file.

4.7.9.4 Domain Data File

Domain data files contain information about a domain, such as the IP addresses and names of the hosts in the domain.

The forward domain data file contains entries that map hostnames to IP addresses for hosts in the domain. The file also contains other data for the hosts, such as mail (MS) or text (TXT) resource records. For dynamic domains with separated static and dynamic data, the forward domain data file, which contains dynamic data, must contain an `$INCLUDE` statement to include the file that contains the static data. For example:

```
$INCLUDE c:\mptn\etc\namedb\static.data static.armonk.cooking.net
```

The reverse domain data file contains entries that map IP addresses to hostnames.

The domain data files can be named as desired, and the chosen name must be put in the NAMED.BT file or specified in the `$INCLUDE` directive.

Figure 99 and Figure 100 on page 146 show the sample forward domain data files for a dynamic secured domain with separated files for dynamic data (DNSF0000.DOM) and static data (DNSF0000.STA).

```

;***** IBM DDNS Server Administrator *****
;This file was written by the IBM DDNS Server Administrator on 28-May-98
;***** IBM DDNS Server Administrator *****
$ORIGIN cooking.net.
armonk      IN      KEY      0x0080  0  1  AQOhHKv3ZhrCc7Hgsq2FglBW0t4qKWv+Gjmf
VQ8sd2lkCVc6NcNv1QNWex21UedaE7AmZSzhKV5FSM9pPqpx8Rkd ;Cl=3
              IN      NS      merlot.armonk.cooking.net.      ;Cl=3
              IN      SOA      merlot.armonk.cooking.net. administrator.armonk.coo
king.net. (
              12 10800 3600 604800 86400 )      ;Cl=3
$ORIGIN armonk.cooking.net.
pizza 3600 IN KEY 0x0000 0 1 AQPTXIKlHuTRxQzSS7qUadPndTt17Vcr8blm
Yo/ZQ72/Q+nkpwoia5NNLmB66G0FVJh62IOYAP8QQ7oZdq0lNMWv ;Cr=auth
4660 IN A 192.168.6.12 ;Cr=auth
4660 IN SIG A 1 4 4660 896410050 896323650 0xc534 pizza.armonk.
cooking.net RgZRe5FQ6Fv3vygft0Xrtq1lZqTSi3d8fw/qjDLqHBLWMel62YRo8xqmuALY8IOHX/ jKCpF
aEBFq9wgt5SfCxA== 896460438 ;Cr=auth
4660 IN SIG KEY 1 4 4660 899520450 896323650 0xc534 pizza.armon
k.cooking.net Kd+Y7p92VvGy5XD5TYHmEwvc5BknB6xd/9G1AAIqFk2NpXewSJSeW4/anLgz2xT6tT3m5
dEsjjgbB6K8/VNbrQ== 899570838 ;Cr=auth
burger 3600 IN KEY 0x0000 0 1 AQOyn4i0WzIt6io3CQOaT7kbqLuFAMBLAhY7
BN7nNnQgy5DCmXD2lac386FgwGds4VwzI6RBYZmDUFyMZXTJSJSH ;Cr=auth
4660 IN A 192.168.6.13 ;Cr=auth
4660 IN SIG A 1 4 4660 896502341 896459141 0x9448 burger.armonk
.cooking.net ebko2wARtFhFxPMEm/Yb69+BPcmx6vQ2hqbfwrlmtmDAspnraXGVVX+V17PRPQoxWlYypk
0UOMhwL5Pj9TX05A== 896495156 ;Cr=auth
4660 IN SIG KEY 1 4 4660 899612741 896459141 0x9448 burger.armon
nk.cooking.net PaJ4+IBi+vYBclxyBsC6WQRN0sxreKECercFCjZFyJEvCpBARDR05nlNvdTB0z20/2oed
+8bMc66cmBRlw9Zo4w== 899605556 ;Cr=auth
$INCLUDE C:\MPTN\ETC\namedb\dnstf0000.sta armonk.cooking.net.

```

Figure 99. [Warp Server] DNSF0000.DOM File

Note: Lines that are longer than 80 characters are wrapped to the next line.

```

;***** IBM DDNS Server Administrator *****
;This file was written by the IBM DDNS Server Administrator on 28-May-98
;***** IBM DDNS Server Administrator *****
$ORIGIN armonk.cooking.net.
ns-updates IN CNAME merlot.armonk.cooking.net.
merlot IN A 192.168.6.10

```

Figure 100. [Warp Server] DNSF0000.STA File

4.7.10 Providing Mail Services

The following discussion pertains to installing mail services on an OS/2 Warp Server using Microsoft and IBM mail clients. The DHCP options and DNS MX record discussion are applicable to any server platform. Only GUI use is platform specific.

Now, we have automatic IP addresses, and our Dynamic DNS server is keeping track of names for us. One of the first tools to set up now is the mail system. We are going to set up mail services on our OS2 Warp Server. There

is a very robust POP3 mail server available called OS2POPS. This was written by IBM Employees and is available for free distribution and use. We have included version 2.02 on the CD-ROM, but for the latest version, see:

<http://www.raleigh.ibm.com/misc/os2pops>

The SMTP mail server (sendmail) is shipped with OS/2 already; so we can have a mail system for no extra money. If you have decided to use Domino, you will probably want to use the *SMTP* (Simple Mail Transport Protocol) and *POP3* (Post Office Protocol) mail servers included in that product (refer to your Domino documentation for that). The DHCP and DNS records we will talk about will be applicable to your Domino installation, as well as the mail system we will install.

What we will show you in this section is how to configure a mail server so that it works with your existing DNS system. You will learn how to install, configure, and support the mail system. We will use the DHCP GUI to provide our mail server address to our clients. We will also use the nslookup tool to examine our name server records. Right now, if you do an nslookup on your name server you will not see any MX (mail exchanger) records. We will add the MX record after we setup the mail server. As mentioned in 4.4.12, "Record Types" on page 99, an MX record indicates a host that will deliver the mail to the addressee, or it will forward the mail to another host that can do the same (that is, deliver or forward). The forwarding is accomplished via SMTP. To summarize, the steps necessary are:

- Install, configure, and run a POP3 mailer.
- Install, configure, and run an SMTP mailer.
- Configure the DDNS server for mail processing or forwarding (MX records).
- Configure DHCP server to provide mailer addresses to clients (options 69, 70).

Installing OS2POPS is simple. Just unzip into the directory where you want it installed, backup \MPTN\ETC\SENDMAIL.CF, and then run the program OS2POPS. You will get a screen similar to that shown in Figure 101 on page 148.

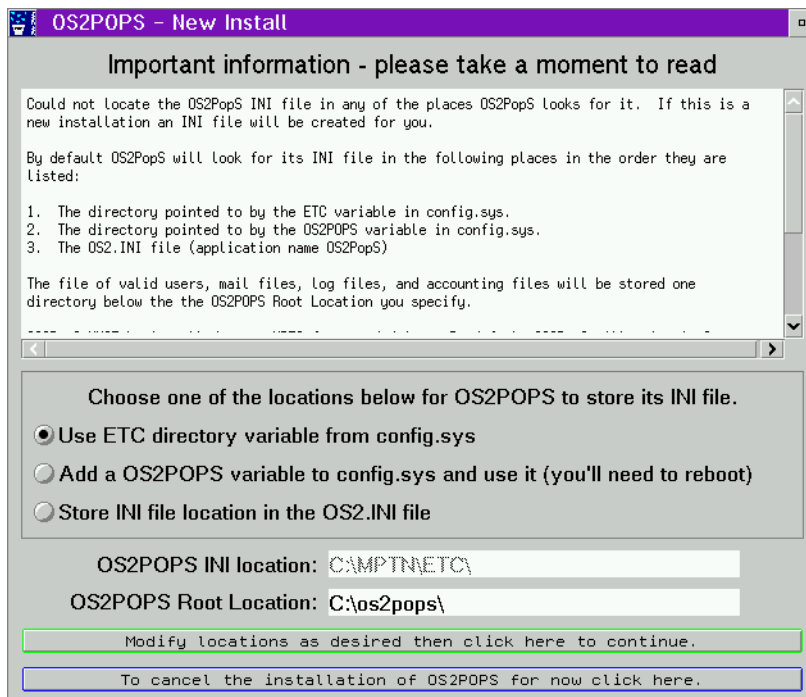


Figure 101. [OS2POPS] Installation Panel

All the OS2POPS files should be in the **OS2POPS Root Location**: shown on the screen. After OS2POPS installs, you can configure mail users by clicking on **Client Maintenance**, as shown in Figure 102.

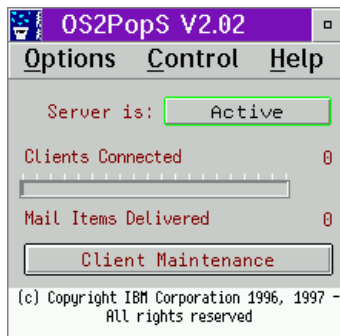
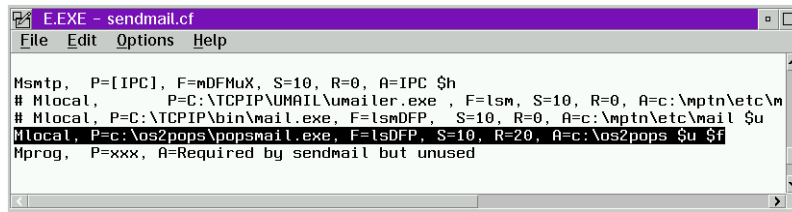


Figure 102. [OS2POPS] Client Maintenance

When OS2POPS installs, it is supposed to modify the C:\MPTN\ETC\SENDMAIL.CF file, but ours did not. We manually modified the Mlocal entry using the E editor as shown in Figure 103 on page 149.



```
Msmtpl, P=[IPC], F=mdfMuX, S=10, R=0, A=IPC $h
# Mlocal, P=C:\TCPIP\UMAIL\umailer.exe, F=lsn, S=10, R=0, A=c:\mptn\etc\m
# Mlocal, P=C:\TCPIP\bin\mail.exe, F=lsmdfP, S=10, R=0, A=c:\mptn\etc\mail $u
Mlocal, P=c:\os2pops\popmail.exe, F=lsdFP, S=10, R=20, A=c:\os2pops $u $f
Mprog, P=xxx, A=Required by sendmail but unused
```

Figure 103. [OS2POPS] Manually Modifying SENDMAIL.CF

Those are single spaces between each entry, not tab characters. Now, the POP3 mailer is installed and working.

E Editor

Be sure to use the E editor, which will leave tab characters alone. You can easily break your C:\MPTN\ETC\SENDMAIL.CF file by using an editor that converts tabs to spaces. Sendmail depends on tabs in certain lines for proper parsing.

We still need to configure the SMTP mailer. First, start the TCP/IP Configuration notebook. Then, go to the Autostart tab and set sendmail to autostart, as shown in Figure 104 on page 150.

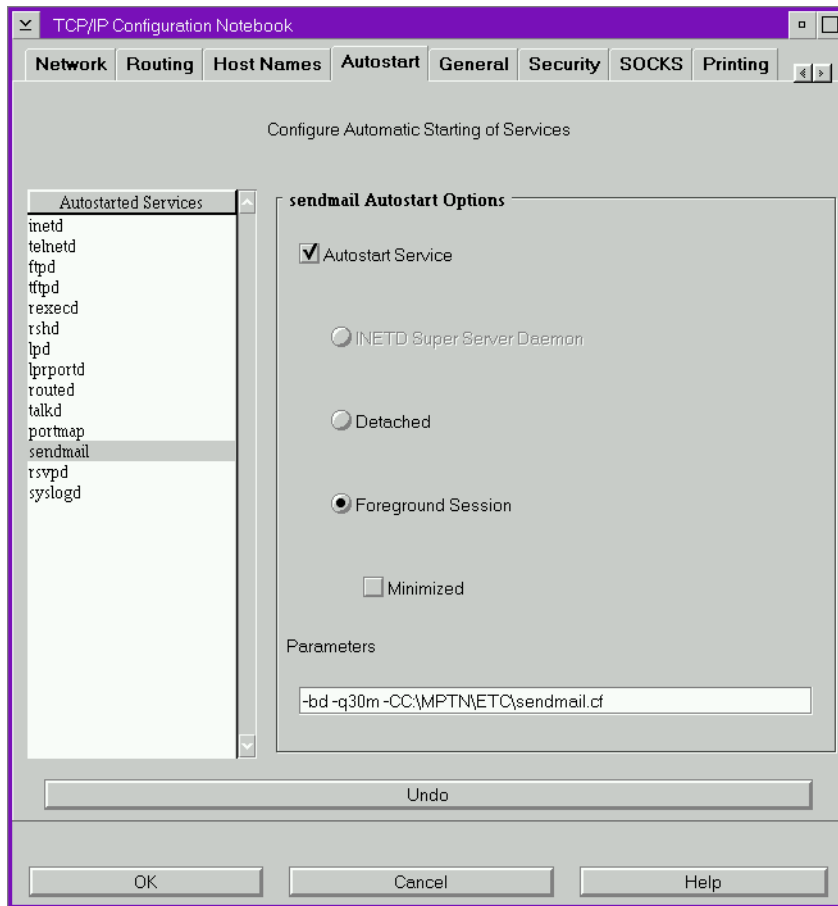


Figure 104. [Warp Server] Setting Sendmail to Autostart

Also notice that we changed the Parameters entry from the default. It was SENDMAIL.UML and we changed it to SENDMAIL.CF. Configuring a sendmail file can be a career itself, and we are not going to describe it any further than the `Mlocal` change we did previously. The SENDMAIL.CF file is as shipped from IBM except for the `Mlocal` change. Click on **OK** to save, and do not reboot at this time (that is, choose **Cancel** when asked if you want to reboot now).

Next, we will update the DHCP server to tell our clients the IP address of the POP3 and SMTP mailers. Start the **DHCP Configuration** GUI and double-click on the global entry, as shown in Figure 105 on page 151.

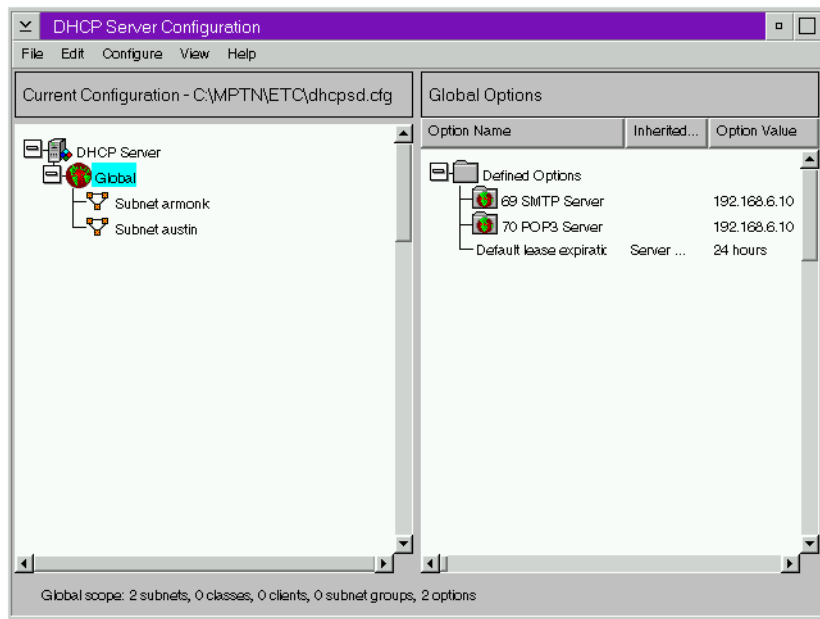


Figure 105. [Warp Server] Modifying Global DHCP Parameters

Choose the **DHCP Options** tab, choose option **69 SMTP Server**, and **Add** the IP address of your server, as shown in Figure 106 on page 152.

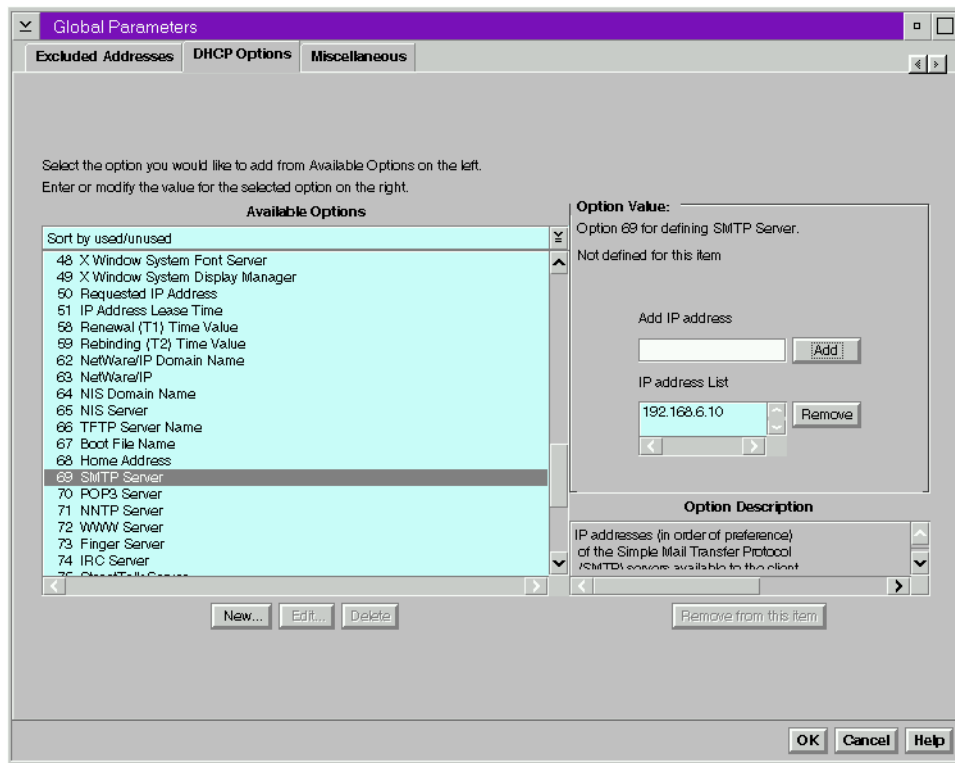


Figure 106. [Warp Server] Defining an SMTP Server

Then, choose option **70 POP3 Server** and **Add** the IP address, as shown in Figure 107 on page 153.

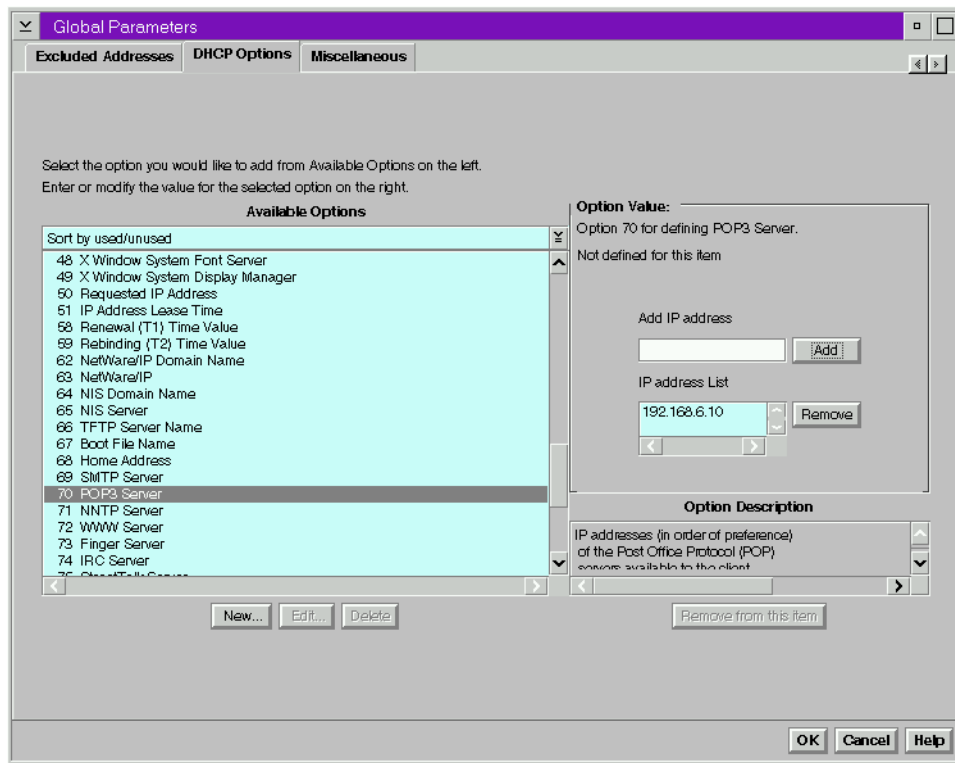


Figure 107. [Warp Server] Defining a POP3 Server

You see that our mail servers are both on the same machine as our DHCP/DDNS server, but there is no requirement for this. If we had a dedicated mail server on our network, we would have used that machine's IP address instead. Click **OK** to save your configuration, and then choose **[File — Save]** to save the new DHCP configuration. Re-initialize the DHCP server using the `DADMIN -I` command.

The penultimate task will be updating our DDNS server with an MX record. Start the **DDNS Administration** GUI and double-click on the **Primary Domain**. Right click the arrow to show the hidden tabs and choose the **Mail Exchange** tab. As shown in Figure 108 on page 154, we chose a preference number of 10 for our mail server. Note that we use the full canonical name for our mail server. Because of the way sendmail function, you are usually safer using a full canonical name for the mail server. See the O'Reilly publication, *Sendmail*, for more in-depth information about sendmail.

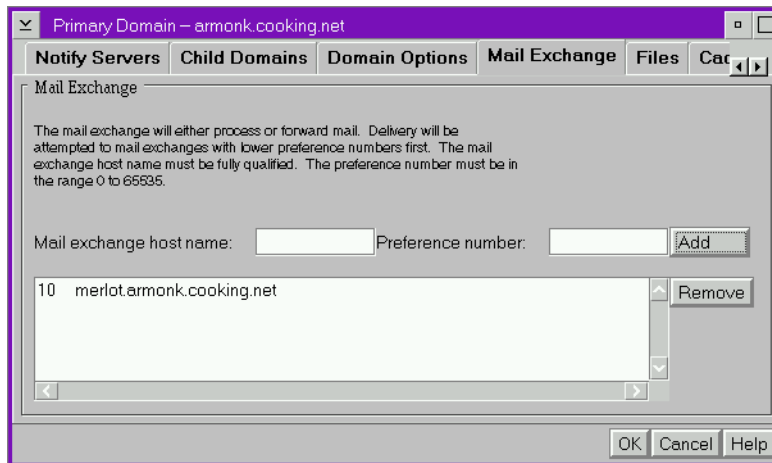


Figure 108. [Warp Server] Adding a Mail Exchange to the Primary Domain

The preference number is used to direct the mailer to the best server for performing mail duties. A lower number for the preference value is better, with zero (0) being the best. The numbers are relative values to each other, not absolute. It is typical to separate numbers by 5 or 10 so that you can insert test mailers without changing any other preference numbers. When we saved our configuration via the GUI, the domain file was modified, as in Figure 109 on page 155. You can see the new MX record with value 10. If we had that dedicated mail server mentioned earlier, we may have entered two servers, the dedicated one at a value of 10, and our Warp Server at a value of 20. This would have caused the dedicated server to be used first, and OS/2 Warp Server would have been a backup mail server.


```

TEDIT.EXE
Top of File
$ORIGIN cooking.net.
armonk      IN      SOA      merlot.armonk.cooking.net. administrator.armonk.
             9 10800 3600 604800 86400 ) ;Cl=3
             IN      KEY      0x0080 0 1 AQ0cJPsqMdBXtAsPmCfL5nTbpqydXY91L
             IN      NS       merlot.armonk.cooking.net. ;Cl=3
             IN      MX       10 merlot.armonk.cooking.net. ;Cl=3
$ORIGIN armonk.cooking.net.
salmon 4660 IN SIG KEY 1 4 4660 898902959 895706159 0x5485 salmon.a
         4660 IN SIG A 1 4 4660 895709908 895706308 0x5485 salmon.arm
         3600 IN KEY 0x0000 0 1 AQPAVMmk0/z0v9sb5v5/nEGHDCEKng03W
burger 4660 IN A 192.168.6.12 ;Cl=3
         4660 IN SIG A 1 4 4660 895761650 895675250 0xa316 BURGER.arm
         4660 IN SIG KEY 1 4 4660 898872050 895675250 0xa316 BURGER.a
         3600 IN KEY 0x0000 0 1 AQPILDDBU0Zjqd5VA2eVDX9MjherKQ1E4
shrimp 4660 IN A 192.168.6.15 ;Cl=3
         4660 IN SIG A 1 4 4660 895792032 895705632 0xc78a shrimp.arm
         4660 IN SIG KEY 1 4 4660 898902432 895705632 0xc78a shrimp.a
         3600 IN KEY 0x0000 0 1 AQ0mJnj0qnnqcRwE36116yArrDptWf5Q2
brie 4660 IN A 192.168.6.12 ;Cl=3
         4660 IN SIG A 1 4 4660 895767085 895680685 0xa316 brie.armon
         4660 IN SIG KEY 1 4 4660 898877485 895680685 0xa316 brie.arm
dnsf0000.dns
F1=Help F2=Save F3=Quit F4=File F5=Cmd F7=Name F8=Edit F9=Undo F10=Next

```

Figure 109. [Warp Server] Modification to Domain File Showing New MX Record

The term best server is arbitrary at best, although generally it refers to mail servers that have a higher capacity or throughput, or a higher bandwidth connection to where most mail is coming or going. The mailer will attempt to use lower numbered servers before it attempts to use the higher numbered mail servers for each mail transaction. If a server is unavailable, extra network traffic is created. This is one of the reasons to make sure the more reliable mail servers are numbered lower (of course, user satisfaction is another reason). The SMTP port number used is 25 and the POP3 port number is 110, as shown in Figure 110 on page 156. This shows the partial output of the `NETSTAT -S` command on our OS/2 Warp Server box.

netstat.exe

[C:\mpn\etc\namedb]netstat -s

AF_INET Address Family:
Total Number of sockets 28

SOCK	TYPE	FOREIGN PORT	LOCAL PORT	FOREIGN HOST	STATE
1	DGRAM	0	netbios-dgm..138	0.0.0.0	UDP
2	DGRAM	0	netbios-ns..137	0.0.0.0	UDP
3	STREAM	0	netbios-ssn..139	0.0.0.0	LISTEN
4	DGRAM	0	emfis-data..140	0.0.0.0	UDP
5	DGRAM	0	tftp..69	0.0.0.0	UDP
6	STREAM	0	telnet..23	0.0.0.0	LISTEN
7	STREAM	0	exec..512	0.0.0.0	LISTEN
8	STREAM	0	ftp..21	0.0.0.0	LISTEN
9	DGRAM	0	sunrpc..111	0.0.0.0	UDP
10	STREAM	0	sunrpc..111	0.0.0.0	LISTEN
11	DGRAM	0	0	0.0.0.0	UDP
12	STREAM	0	pop3..110	0.0.0.0	LISTEN
13	STREAM	0	7464	0.0.0.0	LISTEN
14	STREAM	0	7463	0.0.0.0	LISTEN
15	STREAM	7463	49152	192.168.6.10	ESTABLISH
16	STREAM	49152	7463	192.168.6.10	ESTABLISH
17	STREAM	0	7462	0.0.0.0	LISTEN
18	STREAM	0	0	0.0.0.0	CLOSED
19	DGRAM	0	13991	0.0.0.0	UDP
2066	DGRAM	0	syslog..514	0.0.0.0	UDP
2067	STREAM	0	smtp..25	0.0.0.0	LISTEN
2237	STREAM	0	domain..53	0.0.0.0	LISTEN
2238	DGRAM	0	domain..53	0.0.0.0	UDP
2239	DGRAM	0	domain..53	0.0.0.0	UDP
2241	DGRAM	0	domain..53	0.0.0.0	UDP
2276	DGRAM	0	bootps..67	0.0.0.0	UDP
2277	STREAM	0	942	0.0.0.0	LISTEN
2396	DGRAM	0	0	0.0.0.0	UDP

AF_OS2 Address Family:
Total Number of sockets 6

Figure 110. [Warp Server] NETSTAT -s Showing SMPT and POP3 Sockets

Our final task, as with any system change, is to test our new configuration. We added two users (brie@merlot.armonk.cooking.net and oyster@merlot.armonk.cooking.net) to OS2POPS using the Client Maintenance as described above. Figure 111 on page 157 shows the configuration we used on brie, a Windows 95 machine using Netscape Communicator Version 4.05.

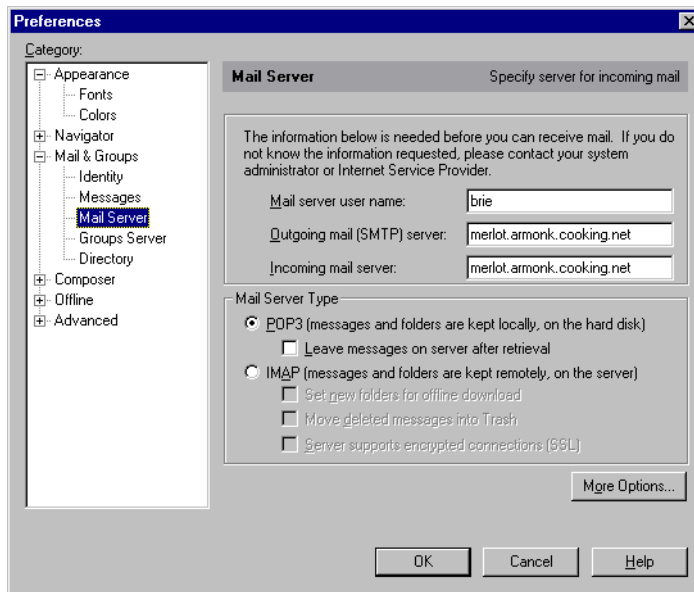


Figure 111. [Netscape] Mail Server Configuration

This is representative of Netscape mail setup on any machine. There are techniques, described in Chapter 7, “Mobile Users” on page 279, for automatically configuring Netscape on client machines. You may be able to modify the steps for automatic client mail configuration. Figure 112 shows oyster responding to its first email from brie.

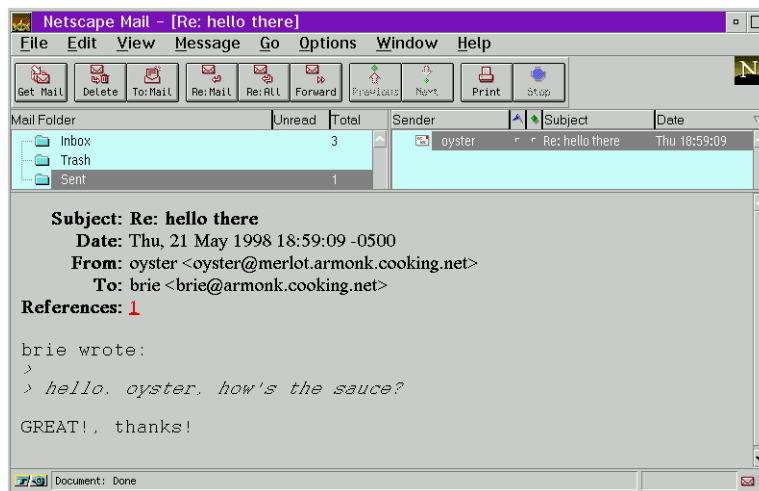


Figure 112. [Netscape] Reading Email

4.8 Dynamic DNS Client Support in OS/2 Warp Server

The following section describes how to enable common clients to work with Dynamic DNS.

4.8.1 OS/2 Warp 4 with TCP/IP Version 4.1 for OS/2

The following steps show you how to set up the DDNS client on OS/2 Warp 4 using TCP/IP Version 4.1 for OS/2.

Enabling DDNS

Follow the steps below to enable the DDNS client:

1. Open the **TCP/IP Shadows** folder and the **TCP/IP Configurations** folder and then double click on **TCP/IP Configuration (Local)** icon or just type `TCPCFG2` at an OS/2 command prompt. The TCP/IP configuration notebook appears.
2. Click on the **Network** tab. Select the **Automatically, Using DHCP** radio button and check **Also, Using DDNS**.

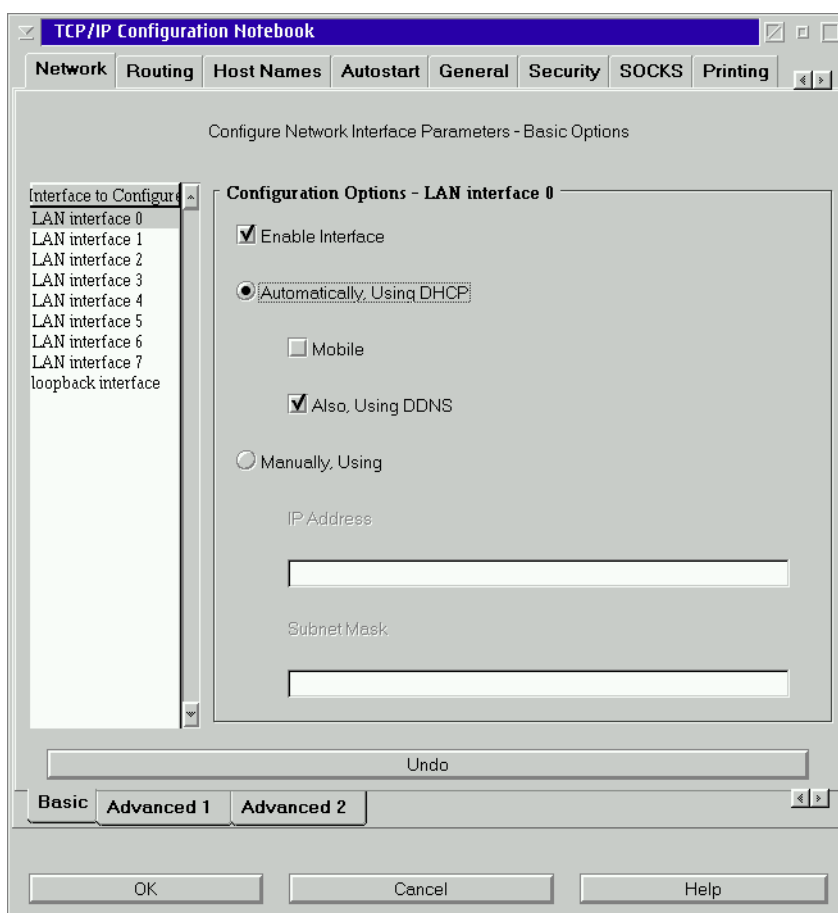


Figure 113. [OS/2 Warp] Enabling the DDNS Client

3. Click on **OK** to save the configuration
4. Reboot your computer.

Configuring the DDNS Client

The DDNS client can be configured using **DDNS Client Configuration** program (DDNSCFG.EXE). You can either type DDNSCFG at an OS/2 command prompt or double-click on **DDNS Configuration** in the System Setup folder to start the program.

When the client is initialized for the first time after DDNS has been enabled, the DDNS Client Configuration window starts automatically and will be displayed. Type your hostname. Also type your domain name and DDNS

Server (they are set by default) if needed. Click **Configure** to update the DDNS server.

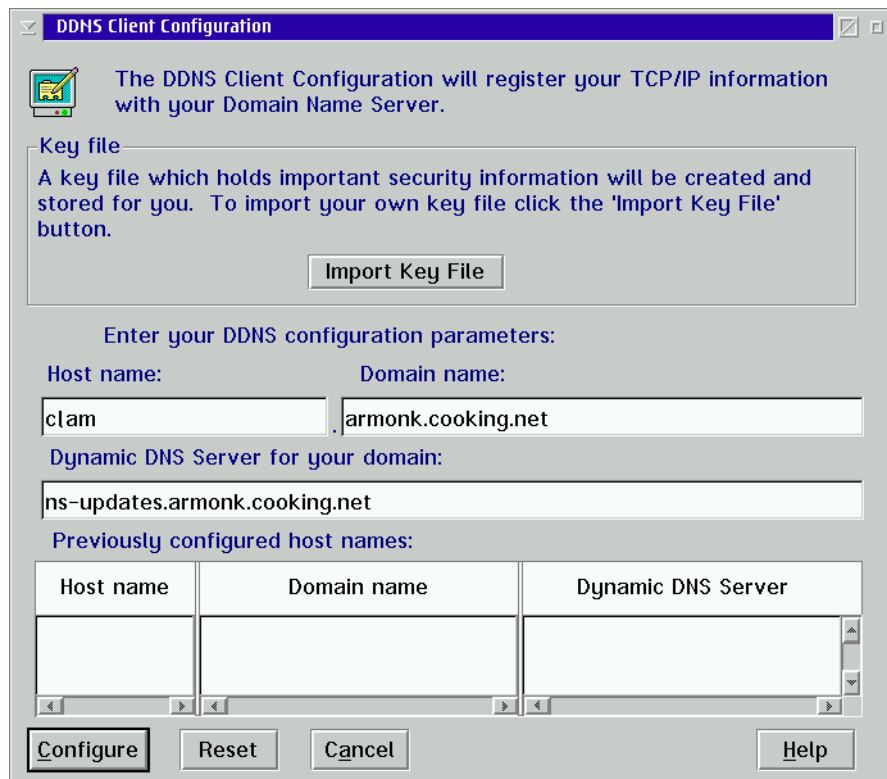


Figure 114. [OS/2 Warp] DDNS Client Configuration

From the next time you start the DDNS Client Configuration program (DDNSCFG.EXE), previously configured hostnames are read-in from the keyfile and displayed for the user to select. Figure 115 on page 161 shows an example.

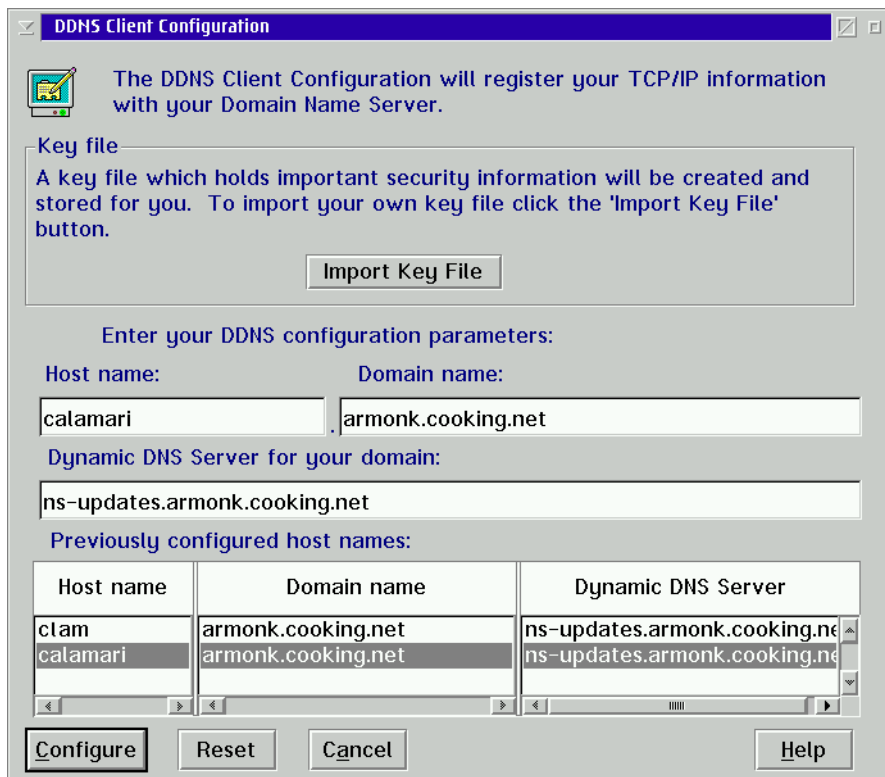


Figure 115. [OS/2 Warp] Selecting a Previous Configuration

Once registered, the DHCP Client Monitor displays the hostname. To check the hostname, double-click on the **DHCP Monitor** object in the System Setup folder.

4.8.2 OS/2 Warp 4 with TCP/IP 4.0

The following steps show you how to set up DDNS client on OS/2 Warp 4 using the TCP/IP version that comes with the product, which is TCP/IP V4.0.

Enabling DDNS

Follow the steps below to enable DDNS client:

1. Open the **Programs** folder and **TCP/IP Internet (LAN)** and then double-click on the **TCP/IP Configuration (LAN)** or just type `TCPCFG` at an OS/2 command prompt. The TCP/IP configuration notebook appears.
2. Select the radio button that says **Automatically, Using DHCP** and mark **Also, Using DDNS** check box.

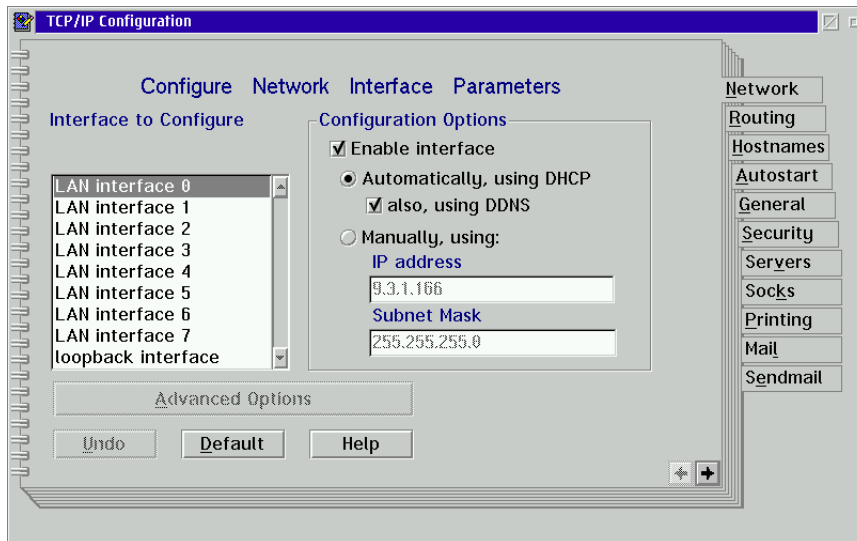


Figure 116. [OS/2 Warp] Enabling the DDNS Client

3. Exit the configuration notebook by closing the window *and* saving the changes.
4. Reboot your computer.

Configuring DDNS Client

The DDNS client can be configured using the DDNS Client Configuration program. You can either type `DDNSCFG` at an OS/2 command prompt or double-click on **DDNS Configuration** in the *System Setup* folder to start the program.

When the client is initialized for the first time after DDNS has been enabled, the DDNS Client Configuration window starts automatically and appears. Type your hostname. Also type your domain name and DDNS server (they are set by default) if needed. Click **Configure** to update the DDNS server.

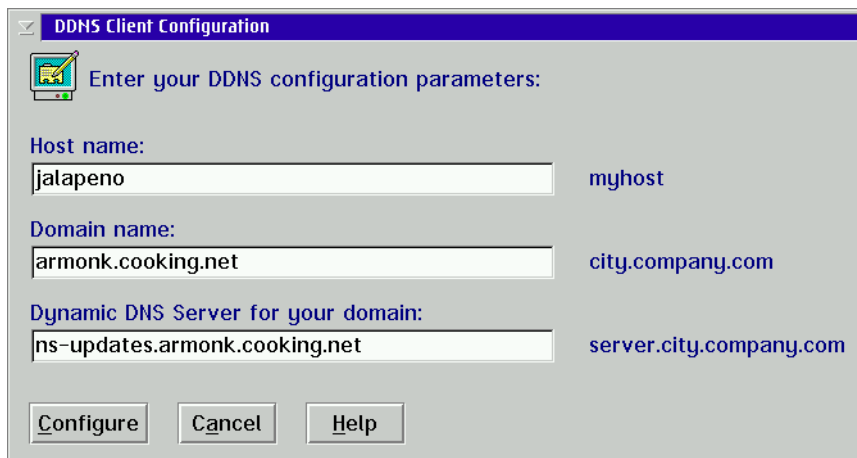


Figure 117. [OS/2 Warp] DDNS Client Configuration

Once registered, the DHCP Client Monitor displays the hostname. Double-click on the **DHCP Monitor** icon in the System Setup folder.

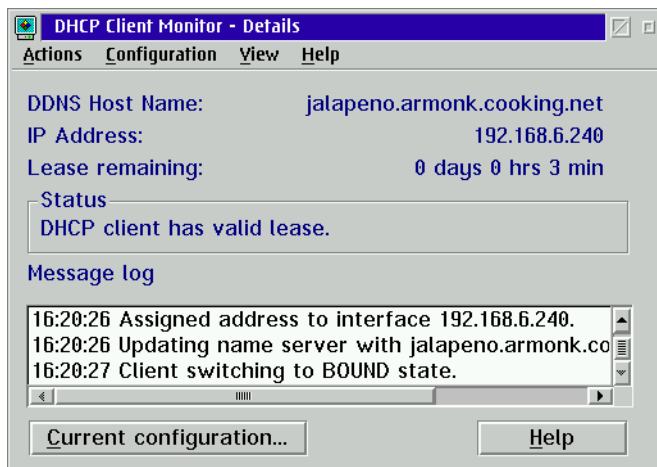


Figure 118. [OS/2 Warp] DHCP Monitor

4.8.3 Windows 95/NT with IBM Dynamic IP Client

The Dynamic IP Client for Windows 95 / NT allows a user to have a hostname mapped to their dynamically allocated IP address given by a DHCP server. This is used in conjunction with the Microsoft DHCP client.

To use the IBM Dynamic Client for Windows 95 / NT, the following items are required:

- Microsoft Windows 95 or Windows NT (Version 4.0 and higher) configured with the TCP/IP protocol
- IBM DDNS Server for OS/2 or AIX
- DHCP Server (Clients must obtain its IP address using DHCP)

The following sections show you how to set up IBM Dynamic IP Client for Windows NT and Windows 95.

4.8.3.1 Windows NT

To install and run the IBM Dynamic IP Client, the user logged in must be a user who belongs to the Administrator Group. This is because only the administrator may manually request/renew the lease or set/configure DHCP information.

Before you configure DDNS client, make sure that DHCP is enabled on that computer. See 2.4.3, "Windows NT Workstation 4.0" on page 46 for more information.

To install the IBM Dynamic IP client, follow the steps below:

1. Open a command prompt. Change to the directory where IBMDYNIP.EXE is located and run the program. Or just double-click on IBMDYNIP.EXE from the Windows NT Explorer.
2. Follow the instructions on each panel to finish installation. When the installation is finished, a message saying *Installation is finished. The IBM Dynamic IP Client Configuration Process will begin* appears. Click **OK**.
3. You can configure your DDNS client at this time or later. If you already know all the information, type the hostname, domain name, and primary domain name server (or just verify the current configuration) and click **OK**. If you wish to configure it later, click **Cancel**.
4. Restart your computer.

If you did not configure the DDNS client when installing IBM IP Dynamic Client, or if you want to change the configuration, follow the steps below:

1. Select [**Programs — IBM Dynamic IP Client — Dynamic IP Configuration**].
2. Type your hostname, domain name, and primary domain name server and then click **OK**.

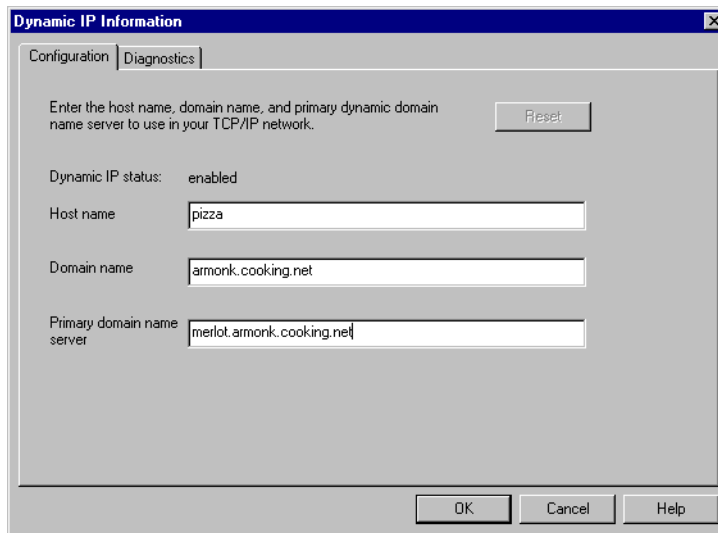


Figure 119. [Windows NT] Dynamic IP Client Configuration

3. The DDNS update runs automatically. If the hostname update was successful, an information window prompts you that the update was successful.
4. To check the return code of the NSUPDATE, open the **Dynamic IP Configuration** and click on the **Diagnostic** tab, as shown in Figure 119.

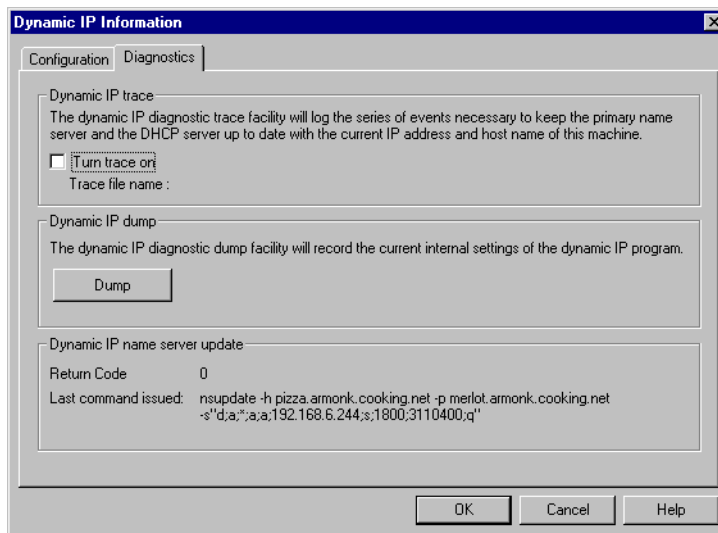


Figure 120. [Windows NT] Dynamic IP Client Diagnostics Tab

5. To check the current configuration, type `IPCONFIG /ALL` at the command prompt.

DDNS updates occur automatically when a user manually renews or releases the lease, using `IPCONFIG.EXE`, or when a user changes the client's hostname, using the Dynamic Configuration program.

4.8.3.2 Windows 95

Before you install and configure the DDNS client, make sure that DHCP is enabled on that computer. See 2.4.2, "Windows 95" on page 45 for how to configure DHCP client.

To install the IBM Dynamic IP Client, follow the steps below:

1. Open a command prompt. Change to the directory where `IBMDYNIP.EXE` is located and run the program. Or just double-click on `IBMDYNIP.EXE` from the Windows Explorer.
2. Follow the instructions on each panel to finish installation. When the installation is finished, a message saying `Installation is finished. The IBM Dynamic IP Client Configuration Process will begin` appears. Click **OK**.
3. You can configure your DDNS client at this time or later. If you already know all the information, type the hostname, domain name and primary domain name server and click **OK**. If you wish to configure it later, click **Cancel**.
4. Restart your computer.

If you did not configure the DDNS client when you installed the IBM IP Dynamic Client, or if you want to make changes to the configuration, follow the steps below:

1. Select [**Programs — IBM Dynamic IP Client — Dynamic IP Configuration**]
2. Type your hostname, domain name, and primary domain name server and then click **OK**.

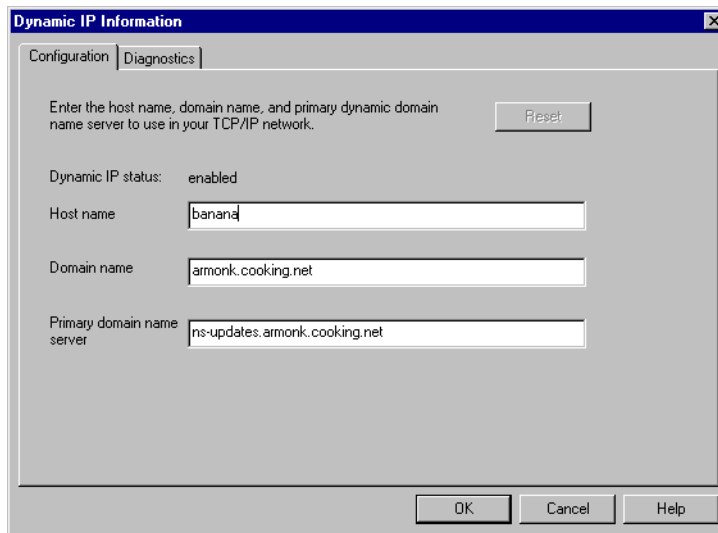


Figure 121. [Windows 95] Dynamic IP Client Configuration

3. Reboot your computer.
4. To check the return code of the NSUPDATE, open the **Dynamic IP Configuration** and click on the **Diagnostic** tab.

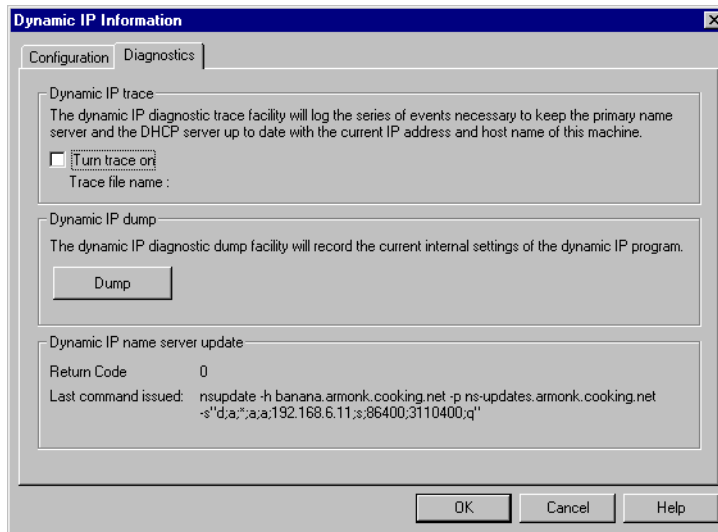


Figure 122. [Windows 95] Dynamic IP Client Diagnostics Tab

5. To check the current configuration, type WINIPCFG at a command prompt.

When running on Windows 95, DDNS updates only occur automatically at the correct DHCP renewal intervals. Manual renewal and release of the lease using `WINIPCFG.EXE` does not coincide with automatic DDNS updates.

When running on Windows 95, you must reboot after changing the configuration of the IBM Dynamic IP Client, as the configuration changes and updates will not occur automatically.

4.8.4 OS/2 Warp 4 Using Presecured Mode

This section explains how to configure an OS/2 Warp 4 DDNS Client using the Dynamic Presecured Domain environment.

Enabling DDNS

Follow the steps below to enable a DDNS client:

1. Open **TCP/IP Shadows** folder and **TCP/IP Configurations** folder and then double click on **TCP/IP Configuration (Local)** icon or just type `TCPCFG2` at an OS/2 command prompt. The TCP/IP configuration notebook appears.
2. Click on the **Network** tab. Select the **Automatically, Using DHCP** radio button and check **Also, Using DDNS**.

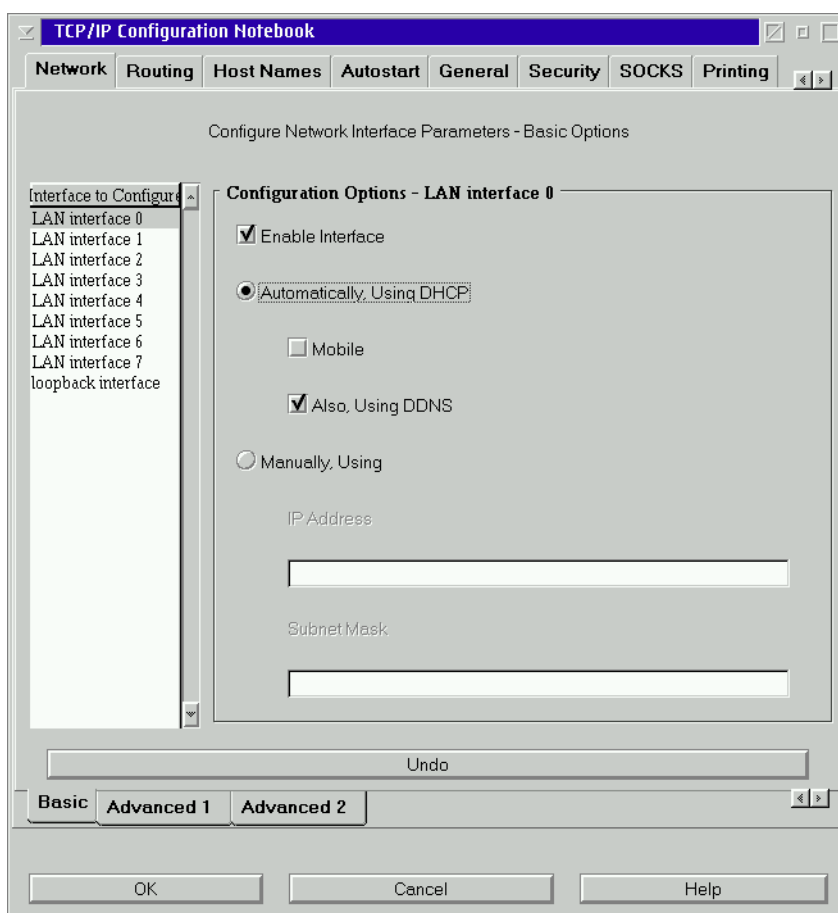


Figure 123. [OS/2 Warp] Enabling the DDNS Client

3. Click **OK** to save the configuration.
4. Reboot your computer.

Configuring DDNS Client

The DDNS client can be configured using the DDNS Client Configuration program. You can either type `DDNSCFG` at an OS/2 command prompt or double-click on the **DDNS Configuration** object in the System Setup folder to start the program.

When the client is initialized for the first time after DDNS has been enabled, the DDNS Client Configuration starts automatically and its window appears. Click **Import Key File** to import the key file to the local machine. Change to

the directory where key file is stored, choose the appropriate key file and click **OK**.

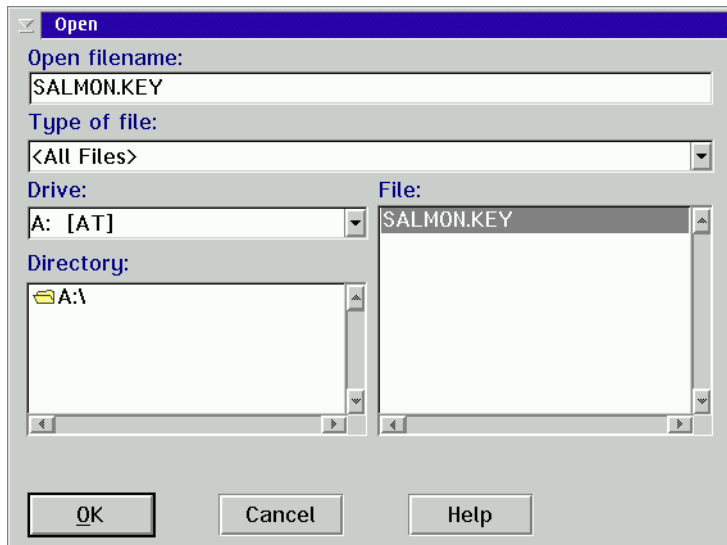


Figure 124. [OS/2 Warp] Importing Key File

If DDNS.DAT file already exists on the local machine, a panel, as shown in Figure 125, appears. You can either replace or append to the existing file.

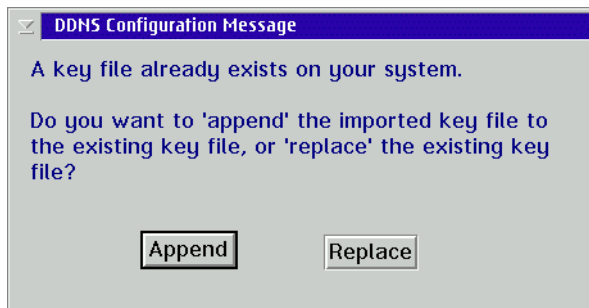


Figure 125. [OS/2 Warp] DDNS Configuration Message

After copying the key file, the hostname, domain name, and Dynamic DNS server name are automatically filled in the DDNS Configuration notebook. Click **Configure** to update the DDNS server.

DDNS Client Configuration

The DDNS Client Configuration will re: with your Domain Name Server.

Key file
 A key file which holds important security information will be created and stored for you. To import your own key file click the 'Import Key File' button.

Enter your DDNS configuration parameters:

Host name: Domain name:

Dynamic DNS Server for your domain:

Previously configured host names:

Host name	Domain name	Dynamic DNS Server
salmon	armonk.cooking.net	merlot.armonk.cooking.net

Figure 126. [OS/2 Warp] DDNS Client Configuration

Once registered, the DHCP Client Monitor displays the hostname. To check the hostname, double-click on the **DHCP Monitor** icon in the System Setup folder.

4.8.5 Windows 95 Using Presecured Mode

This section explains how to configure an IBM Dynamic IP Client on Windows 95 in the Dynamic Presecured Domain environment.

4.8.5.1 Setting Up

Before you install and configure a DDNS client, make sure that DHCP is enabled on that computer. See 2.4.2, "Windows 95" on page 45 on how to configure a DHCP client.

To install the IBM Dynamic IP Client, follow the steps below:

1. Open a command prompt. Change to the directory where IBMDYNIP.EXE is located and run the program. Or just double-click on **IBMDYNIP.EXE** from the Windows Explorer.
2. Follow the instructions on each panel to finish installation. When the installation is finished, a window appears that says *Installation is finished*. The IBM Dynamic IP Client Configuration Process will begin. Click **OK**.
3. You can configure your DDNS client at this time or later. If you already know all required information, type the hostname, domain name, and primary domain name server and click **OK**. If you wish to configure at a later time, click **Cancel**. Do not restart your computer at this time.
4. Copy a key file (<host_name>.key is the default name) in the \WINDOWS\DDNS\SETC directory (\WINDOWS is the directory where Windows 95 is installed) and rename the file name to DDNS.DAT. If you already have a DDNS.DAT file, you can either replace or append the file to the existing file.
5. Restart your computer.

If you did not configure the DDNS client when you installed the IBM IP Dynamic Client, or if you want to make changes to the configuration, follow the steps below:

1. Select [**Start — Programs — IBM Dynamic IP Client**] and then click **Dynamic IP Configuration**.
2. Type the hostname, domain name, and primary domain name server (fully qualified domain name) and then click **OK**. See Figure 121 on page 167.
3. Reboot your computer.
4. To check the return code of the NSUPDATE, open the **Dynamic IP Configuration** and click on the **Diagnostic** tab. See Figure 122 on page 167.
5. To check the current configuration, type WINIPCFG at the command prompt.

When running on Windows 95, DDNS updates only occur automatically at the correct DHCP renewal intervals. Manual renewal and release of the lease using WINIPCFG.EXE does not coincide with an automatic DDNS updates.

When running on Windows 95, you must reboot the machine after changing the configuration of the IBM Dynamic IP Client because the configuration changes and updates will not occur automatically.

Tips for Setting Up

In 4.8.4, "OS/2 Warp 4 Using Presecured Mode" on page 168, we discussed that after importing a key file, the hostname, domain name, and primary DDNS server name were automatically filled in for you in the configuration notebook. However, using IBM's Dynamic IP Client for Windows 95/NT, this information is not filled automatically (because there is no function to import a key file).

The information below describes our actions for a Windows 95 client. We use one Windows 95 computer with IBM Dynamic IP Client installed and made use of its registry information. We exported the registry file, modified the configuration for each one of the other Windows 95 computers, and distributed the appropriate registry file to them:

1. Open the registry.
2. Keys and values to modify are as follows:

HKEY_LOCAL_MACHINE/SOFTWARE/IBMDHCP/CLIENT/1.0/APPLICATION DATA

Set a DDNS server's IP address or a hostname (fully qualified name) in **PrimaryDNS**.

HKEY_LOCAL_MACHINE/SYSTEM/CURRENTCONTROLSET/SERVICES/VXD/MSTCP

1. Set domain name in **Domain**.
2. If **EnableDNS** is set to 0 (which means disabled), set the value to 1.
3. Specify a hostname of the client computer in **Hostname**.
3. Highlight **MSTCP** in the trees and select the **Registry** menu and **Export Registry File**. This will create <file_name>.REG file on the desktop. Exported registry files are text files. You can edit the file using an editor for other clients.
4. Export **IBMDHCP** in the same way.
5. On the target client, double-click on these registry files. This will register the information in the file into the registry.



Figure 127. [Windows 95] Registry Editor Message

6. Then, install the IBM Dynamic IP Client on the target machine. When the configuration notebook comes up at the end of the installation, it already contains all the information needed.

4.8.6 Windows 95 Using Proxy

The DHCP server updates both forward and reverse mappings of the client's hostname and IP address. the only configuration needed on the client computer is to enable a DHCP client. To enable a DHCP client, refer to 2.4.2, "Windows 95" on page 45.

Windows 95 DHCP client sends its computer name (NetBIOS name) as Option 12, and a DHCP server uses this name to update DDNS server.

4.8.7 WorkSpace On-Demand Using Proxy

Since WorkSpace On-Demand Client loads all the software it needs from WorkSpace On-Demand Manager (server), all the configuration is performed at the server.

Configuration

To enable the DDNS client for WorkSpace On-Demand Client, follow the steps below. Also, see 2.4.9, "IBM WorkSpace On-Demand" on page 67.

1. Start the **LAN Server Administration** program and open the **Remote IPL Requesters** folder.
2. Open the **WorkSpace On-Demand Client** icon.
3. Make sure that DHCP client is enabled by checking **Automatic, using DHCP**.
4. Check **Also using DDNS**. By doing this, fields other than the hostname and domain name are grayed-out. Fill in the hostname and domain name. Click **Set** to save and close the notebook.

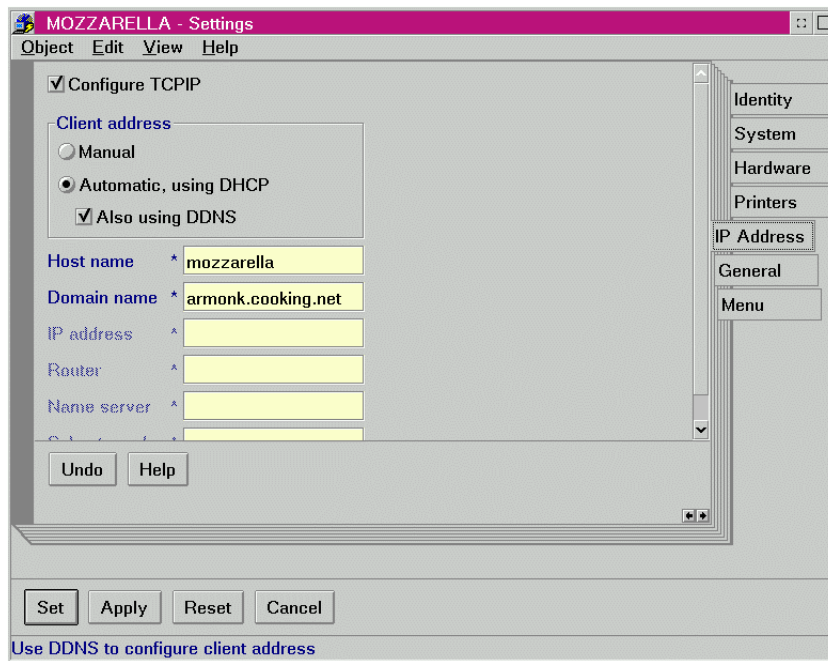


Figure 128. [WSOD] TCP/IP Setting Notebook

The hostname you specified in the WSOD client notebook is added as Option 12 (hostname) and the domain name is added as Option 15 in the DHCP Client Configuration file (DHCPD.CFG). These are sent to a DHCP server.

Start the WSOD client. The WSOD client gets an IP address from the DHCP server and the DHCP server registers the hostname for the WSOD client.

In a regular OS/2 Warp machine, the DDNS Client Configuration program (DDNSCFG.EXE) is launched automatically when the client is initialized for the first time after DDNS has been enabled. The DDNS client starts automatically every time the machine is initialized. However, this does not occur at the WSOD client. Even though we check **Also using DDNS** in the TCP/IP configuration, the WSOD client does not register its hostname by itself, and a ProxyArec-enabled DHCP server would update the DDNS server.

Tip: Delete DDNSCFG.EXE from the RIPL Tree

If you open a DHCP monitor on the WSOD client, you will notice that it says *Registration Failed*, as shown in Figure 129. This is OK. The reason for that is that the monitor only shows a hostname registered by the DDNS Client Configuration program. (DDNSCFG.EXE). It does not show a hostname registration through ProxyArc.

We recommend administrators deleting DDNSCFG.EXE from the RIPL tree so that this program cannot be accidentally executed by the WorkSpace On-Demand user.

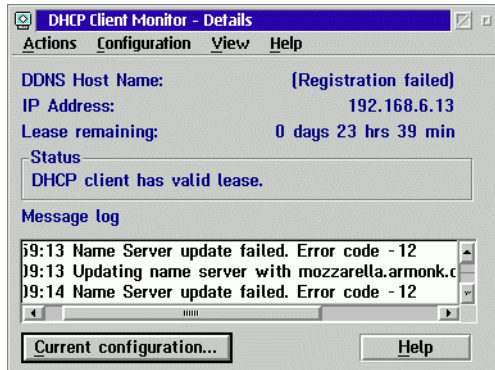


Figure 129. [WSOD] DHCP Client Monitor

4.9 Related Publications

1. *DNS and BIND*, 2nd Edition; O'Reilly & Associates; Albitz and Liu; ISBN: 1-56592-236-0.
Note: A third edition of this book will be available in September 1998.
2. *Sendmail (Nutshell Handbook)*, 2nd Edition (January 1997); O'Reilly & Associates; Costales and Allman; ISBN: 1-56592-222-0.
3. RFC 810 *DoD Internet Host Table Specification*.
4. RFC 1034 *Domain Names - Concepts and Facilities*.
5. RFC 1035 *Domain Names - Implementation and Specification*.

Chapter 5. Integrating File and Print Services

In this chapter, we aim to show how to integrate your existing LANs into your TCP/IP network. This chapter focuses on the "legacy" protocol NetBEUI, NetBIOS name servers, Microsoft WINS and how to enable your clients to use these services. If you are interested in integrating printing and file sharing into your TCP/IP network then you should read this chapter.

5.1 NetBIOS, NetBEUI and TCPBEUI: What Are They?

NetBIOS was originally developed by Sytek Inc. for IBM to link a network operating system with specific hardware. NetBIOS offers LAN applications a variety of "hooks" to carry out inter-application communications and data transfer. Essentially, NetBIOS is a way for application programs to talk to the network.

NetBEUI (pronounced "net-booeey") is the original PC networking protocol and interface designed by IBM for OS/2 LAN Server/OS/2 Warp Server. This protocol was later adopted by Microsoft for their networking products.

NetBEUI is optimized for very high performance when used within a LAN segment, but does not allow for packet forwarding on routed networks. The NetBIOS interface does not rely on any specific underlying protocol to connect systems together, and as such, it is adaptable to protocols that allow packet forwarding, such as TCP/IP.

Seeing that NetBEUI is not a routable protocol, it is a major limitation for a growing company where routable protocols such as TCP/IP are gaining popularity

RFCs 1001 and 1002 describe the standard way to implement the NetBIOS services on top of the Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). RFC 1001/1002 is not an encapsulation technique; it builds special packets and sends them out via UDP/IP and TCP/IP. NetBIOS over TCP/IP is known as *NBT* or *TCPBEUI*.

Remember

TCPBEUI is translation, not encapsulation.

Once a NetBIOS session has been established, TCPBEUI will use sockets-send commands over a TCP connection to send NetBIOS session data. TCPBEUI builds a 4-byte session header that precedes the actual user data. Therefore, a NetBIOS chain send of 128 KB would have an overhead of only four bytes.

IBMs OS/2 Warp and Microsoft Windows NT implement RFCs 1001 and 1002 in order to overcome the limitation of NetBEUI. Figure 130 shows how a workstation can communicate across an IP only router with traditional TCP/IP hosts, as well as with servers using the TCPBEUI protocol. The workstation can continue to communicate to local servers using NetBEUI.

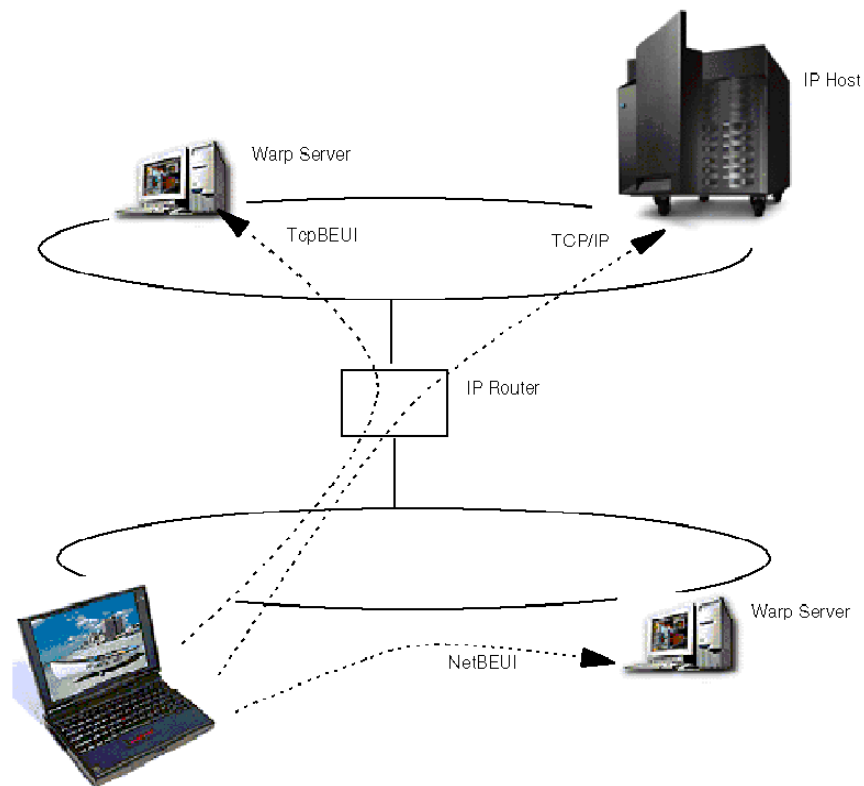


Figure 130. Workstation Communicating Using Multiple Protocols

Figure 131 on page 179 shows the structure of NetBIOS, NetBEUI and TCPBEUI when implemented under OS/2. Data flows within the diagram are vertical only.

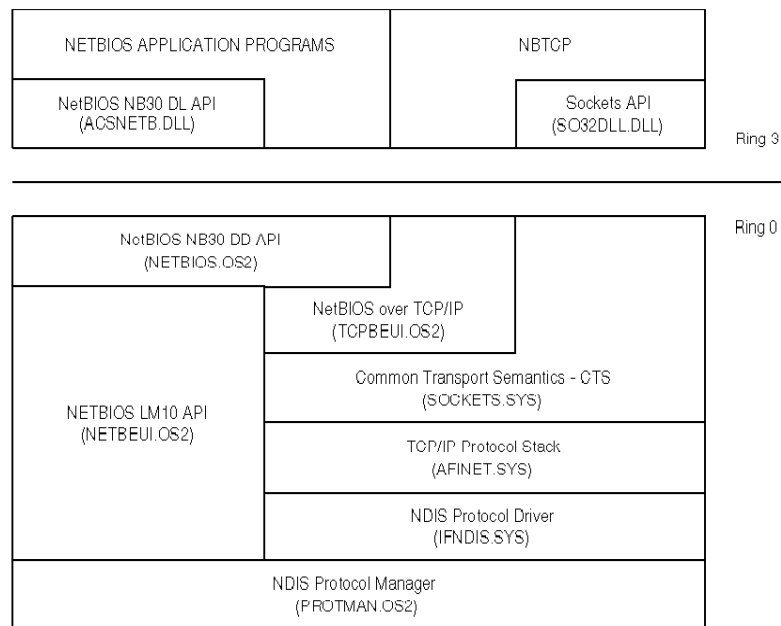


Figure 131. [OS/2 Warp] NetBIOS, NetBEUI and TCPBEUI Structure

Whenever a NetBIOS-based computer (NetBEUI, NetBEUI or TCPBEUI) connects to a network, it registers several NetBIOS names to the network. NetBIOS names are made up of 16 bytes which normally consist of a 15 characters name with a 1 character suffix. All 16 bytes can be binary. These names can be unique to the workstation (for example, the computername), or they can be a group name (for example the workgroup or domain name). Table 8 on page 180 lists the various names and suffixes used.

Table 8. NetBIOS Names and Suffixes

Name	Number ¹	Type ²	Usage
computername	00	U	Workstation Service
computername	01	U	Messenger Service
computername	03	U	Messenger Service used when sending and receiving messages. This is the name registered with the WINS server
computername	05	U	Messenger Service. Used to check if the name registered has been forwarded to another workstation.
computername	06	U	RAS Server service
domain	1B	U	Domain Master Browser, which clients use to contact the domain master browser
domain	1D	U	Master Browser for server lists
computername	1F	U	NetDDE Service
computername	20	U	File Server Service
computername	21	U	RAS Client Service
computername	BE	U	Network Monitor Agent
computername	BF	U	Network Monitor Application
domain	00	G	Domain Name
__MSBROWSE__	01	G	The name master browsers broadcast to announce their domains to other Master Browsers
domain	1C	G	Registered by the Domain Controller, it contains a list of specific addresses of systems that have registered the name.
domain	1E	G	Browsers broadcast and then listen on this name to elect a master browser. This broadcast should be on the local subnet only.
groupname	20	G	A special name registered with WINS servers to identify groups of computers for management purposes.
¹ Numbers are hexadecimal ² U = Unique, G = Group			

To communicate over TCPBEUI, two computers refer to one-another using these names. A problem occurs because they must somehow resolve these NetBIOS names into IP addresses.

5.2 Resolving NetBIOS Names to IP Addresses

When the client connects to the server the client must resolve the server's NetBIOS names. There are various ways to resolve NetBIOS names into IP addresses.

The simplest method is to try broadcasting name queries and hope the computer you're trying to communicate with responds in order to do the resolution. Computers running in this mode are said to be *B-Node* (Broadcast) clients.

A better method would be to have a server on the network that could do the NetBIOS name to IP addresses resolution for us. Such a server is known as a NetBIOS Name Server or *NBNS*. Microsoft uses *WINS* as a type of NBNS. Computers that are configured to use a NBNS are *P-Node* (Peer-to-Peer) clients. Broadcasting is never used by P-Node clients.

Finally, we could combine the modes to become either *M-Node* (Mixed) or *H-Node* (Hybrid) clients.

An M-Node client will attempt a name query broadcast first, and if that fails it will try to use the NBNS.

H-Node clients reverse the sequence used by M-Node clients, and will attempt to use the NBNS first, and if that fails will try a name query broadcast. If an H-Node client detects that a NBNS has failed, it will continue to poll the NBNS (while using broadcasts), so that it knows when to switch back to using the NBNS. H-Node has generally replaced M-Node.

NetBIOS name servers will be covered in Section 5.4, "NetBIOS Name Servers (NBNS)" on page 201. Configuring clients to make use of the NBNS is covered in section 5.5, "P-Node, H-Node and M-Node Clients" on page 209. In the meantime, we'll look at how to enhance the performance of B-Node clients by reducing the amount of broadcasts that they generate.

There is a new way of resolving a NetBIOS name to IP address, not using a broadcast or NetBIOS name server. This approach is discussed in Chapter 5.3, "New Way of Using DNS with Non-RFC-Encoded Name" on page 195.

5.2.1 Enhancing B-Node Clients

Operating as a B-Node client will work for you as long as all the routers and bridges within your network are configured to pass on broadcast traffic. However, generally they're not, so computers will only be able to communicate with one another if they are located on the same subnet

There are enhancements available to B-Node clients in order to work around this limitation.

5.2.1.1 Enhancing DOS, Windows 3.11 and WfW Clients

IBMs DOS LAN Services (*DLS*) comes with a TCP/IP stack from Network TeleSystems that supports DHCP and TCPBEUI. Throughout this chapter when we're discussing DOS, Windows 3.11 or Windows for Workgroups (*WfW*)-based computers, we will assume that DLS is installed.

DLS also includes the PING utility for testing IP connectivity, and a utility called NBUTIL to build a NetBIOS name to IP address resolution table.

The NBUTIL Utility Program

The NBUTIL utility program that comes with DLS allows you to create and maintain a NetBIOS name to IP Address table in memory.

NBUTIL Syntax

```
NBUTIL [-?]
        [-a <ipaddress> <netbiosname>]
        [-f <filename>]
        [-l <number>]
        [-x]
        [-t]
        [-s]
        [-c]
        [-w]
```

Where:

?	Displays a description of all command line options used with NBUTIL.
-a	Adds a NetBIOS name/IP address pair to the NetBIOS name table.
<ipaddress>	The IP address associated with the NetBIOS name you want to add to the NetBIOS name table.

<i><netbiosname></i>	Is the NetBIOS name for the remote system you want to add to the NetBIOS name table.
-c	Clears all entries from the NetBIOS name table.
-f	Specifies the ASCII file containing the NetBIOS names with their associated addresses.
<i><filename></i>	The name of an ASCII text file containing the NetBIOS names with their associated IP addresses.
-l <i><number></i>	This option represents the LAN adapter number (LANABase) used by the TCP/IP protocol driver. If -l is not specified, a default of (0) is used.
-s	Displays all the current entries in the NetBIOS name table.
-t	Enables you to put embedded spaces in NetBios names by translating an underscore(_) into a space. When used with the -s option, -t translates spaces back into underscores.
-w	Translates NetBIOS names into the format of the LAN Manager workstation name.
-x	Translates the NetBIOS name you want to add to the NetBIOS name table into names that can be used by servers running LAN Manager.

Remember

The NetBIOS names table is for NetBIOS to IP address resolution only. It is not used for IP host name to IP address resolution.

Using NBUTIL

The NBUTIL command must be executed after the TCP/IP protocol driver is running; that is, after DLS has started.

You can use the NBUTIL command to add entries to the NetBIOS name table in two different ways:

- By adding the IP address of each system you want to access with its associated NetBIOS name one at a time.
- By creating an ASCII text file containing a number of IP addresses with their associated names, and by using the -f option, all of these names can be added to the NetBIOS name table by invoking the NBUTIL command only once.

The format of the file is:

<IP address> <NetBIOS name> <# comment>

<IP address> Is the IP address of the remote system you want to access.

<NetBIOS name> Is a NetBIOS name associated with the IP address of the remote system you want to access.

<#comment> Is a number symbol (#) followed by any additional information you want to add to the file as a comment.
comment is optional.

```
192.168.6.10 merlot # The MERLOT server
192.168.7.10 texmex # The TEXMEX server
```

Figure 132. [DOS LAN Services] Example Input File for NBUTIL

If you are running DLS under DOS, and want to run the command automatically each time your PC is started, place the command in the AUTOEXEC.BAT file after the NET START command.

```
@ECHO OFF
SET PATH=C:\;C:\NET;C:\DOS
PROMPT $P$G
LH MOUSE
LH SHARE
SET TCPHELP=C:\NET
SET ETCDIR=C:\ETC
C:\NET\NET START
NBUTIL -A 192.168.7.10 texmex -X
```

Figure 133. [DOS LAN Services] Example AUTOEXEC.BAT Using NBUTIL

If you are running DLS under Windows, you must run the NBUTIL program from a DOS box after Windows has been started. This is because the protect mode driver does not load until Windows has started. You must wait until then to run NBUTIL. This can be done automatically by creating an icon to the Startup folder.

Another option is to consider is using LAN logon scripts to centrally manage the names that are in the table.

As a test, we created a user ID on a Warp Server domain that was assigned a Z: drive at logon that contained a file called NBUTIL.LST. (Identical to the

example file shown in Figure 132 on page 184). The user's PROFILE.BAT contained the command

```
C:\NET\NBUTIL -F Z:\NBUTIL.LST -X
```

After logging onto the LAN from either a DOS DLS or Windows DLS workstation, the NBUTIL -S command was run to produce the following output.

```
C:\>NBUTIL -S

NBUTIL - Netbios Name Utility 2.09 (960320)
Copyright (c) 1994-1996 Network TeleSystems, Inc. All rights reserved.
There are 8 entries in the Name/IP cache:

TEXMEX          192.168.7.10
TEXMEX          © 192.168.7.10
TEXMEX          192.168.7.10
TEXMEX          § 192.168.7.10
MERLOT          192.168.6.10
MERLOT          © 192.168.6.10
MERLOT          192.168.6.10
MERLOT          § 192.168.6.10
```

Figure 134. [DOS LAN Services] Sample Output from NBUTIL

You could also load the file using a UNC name. For example:

```
C:\NET\NBUTIL -F \\MERLOT\CONFIGS\NBUTIL.LST -X
```

5.2.1.2 Enhancing Windows 95 and Windows NT Clients

Both Windows 95 and Windows NT can use a file called LMHOSTS to perform NetBIOS name to IP Address resolution.

Under Windows 95, the file should be located in the \WINDOWS subdirectory, while under NT, it is located in \WINNT\SYSTEM32\DRIVERS\ETC. If the file does not exist, a sample file called LMHOSTS.SAM in these subdirectories can be copied to LMHOSTS.

At its simplest, the file contains NetBIOS name and IP address pairs. Additional options are available to enhance the configuration.

LMHOSTS Options

All options should be in upper case.

# <comment text>	Indicates that any text following is a comment unless it is one of the options. '#' is used with all the keywords to maintain backwards compatibility with LAN Manager - which will see the options as comments.
------------------	--

#PRE	Specifies that the address should be preloaded into the name cache. The LMHOSTS file is used only after a WINS query and a broadcast have both failed, and these searches are attempted only if the name is not already in the cache.
#DOM: <domain name>	This option is needed for all servers that validate logon requests. All computers in the domain will need one of these entries for each domain controller in the domain. Note that #PRE must be before #DOM.
#NOFNR	Do not use Directed Name Queries (DNS). Some versions of LAN Manager cannot handle NetBIOS DNS requests and return an error.
#BEGIN_INCLUDE #INCLUDE <other_file> #INCLUDE <other_file> #END_INCLUDE	Include LMHOSTS definitions from other files. The other file may be remote (Remember, the computer must be able to resolve the name of the remote computer before it can use a remote file).
#BEGIN_ALTERNATE #INCLUDE <other_file> #INCLUDE <other_file> #END_ALTERNATE	Defines a list of alternate locations for an additional LMHOSTS file to be included. The first additional file read successfully will be used.
#MH	Associates a NetBIOS name with multiple IP addresses (For example, a server with two or more NICs). The NetBIOS name can be associated with up to a maximum of 25 different IP addresses.

Important

The name cache and the LMHOSTS file are read sequentially, so add the most frequently accessed computer names to the top of the LMHOSTS file, with the #PRE entries at the end of the file. Once entries have been loaded into the cache, they do not need be accessed again in the file.

Figure 135 on page 187 shows a sample LMHOSTS file for a computer that can participate in two domains (REDMOND and BELLVUE). Using the #BEGIN_ALTERNATE option, an additional LMHOSTS configuration file is loaded from either the PIE or HOTDOG server (The first one successfully

loaded will be used). An entry also exists for a DOS-based workstation called BURGER.

Notice that the server entries are located before the #BEGIN_ALTERNATE entries and they are pre-loaded. This allows the server names in the alternate entries to be successfully resolved without having to refer back to the LMHOSTS file.

```
192.168.6.13 BURGER # DOS workstation

192.168.8.10 PIE      #DOM:REDMOND #PRE # PDC for REDMOND domain
192.168.7.9  HOTDOG  #DOM:BELLVUE #PRE # PDC for BELLVUE domain

#BEGIN_ALTERNATE # Load further configuration file
#INCLUDE \\PIE\\CONFIGS\\LMHOSTS
#INCLUDE \\HOTDOG\\CONFIGS\\LMHOSTS
#END_ALTERNATE
```

Figure 135. [Windows 95/NT] Sample LMHOSTS File

With Windows 95, the LMHOSTS file must exist to be used by the operating system. With Windows NT, use of LMHOSTS must be explicitly turned on by selecting the **Enable LMHOSTS Lookup** checkbox as shown in Figure 136.

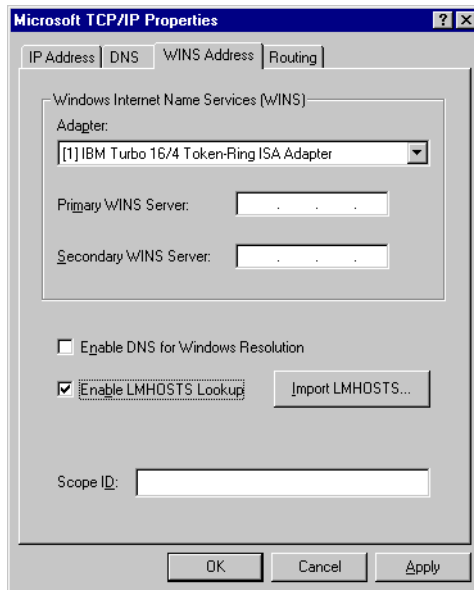


Figure 136. [Windows NT] Enabling Use of LMHOSTS

Windows 95 computers may encounter problems when trying to `NET VIEW` a domain that is on the other side of a router if the computer is only using an LMHOSTS file to do NetBIOS names resolution and not a NBNS. It is necessary to add the following to the LMHOSTS file.

```
<IP address> <tab> "<domain_name> \0x1B" <tab> #PRE
```

<IP address> The IP address of the Primary Domain Controller.

<tab> A TAB character.

<domain_name> The name of the domain. If the Domain name is less than 15 characters, use spaces to pad it to 15 characters. The hexadecimal value 1B is appended following the Domain name as the 16th character (A total of 20 characters including the "\0x1B"). The Domain name must be all upper-case letters and must be inside quotation marks.

5.2.1.3 Enhancing OS/2 Warp Clients

Under OS/2 Warp, there are three enhancements available to reduce the amount of name query broadcasts generated by computers using TCPBEUI. Two of these require configuring text files on each computer, while the other makes use of an existing DNS on the network.

Names File Routing Extension

This routing extension is implemented through the use of a names file, which contains NetBIOS name and IP address pairs. The names file, called RFCNAMES.LST, is located in the \IBMCOM directory.

This implementation requires an entry in the names file for every workstation with which you want to communicate. This file is conceptually similar to the TCP/IP hosts file.

Remember

The RFCNAMES.LST file is for NetBIOS name to IP address resolution only. It is not used for IP host name to IP address resolution.

The names file routing extension is enabled by setting the NAMESFILE parameter in the PROTOCOL.INI file equal to a nonzero integer value, which represents the maximum number of entries in the names file.

The RFCNAMES.LST format is:

```
<string> <IP address>
```

Where:

- <string> A string of characters used when searching for NetBIOS names in the file, delimited by double quotation marks.
- <IP address> Either an IP address in dotted decimal form or a host name string that occurs in the local hosts file or on an Internet domain name server.

The RFCNAMES.LST file can be modified manually, or by using MPTS and selecting the **Names list** option when you **Edit** configuring **IBM OS/2 NetBIOS OVER TCP/IP**. As shown in Figure 137 and Figure 138.

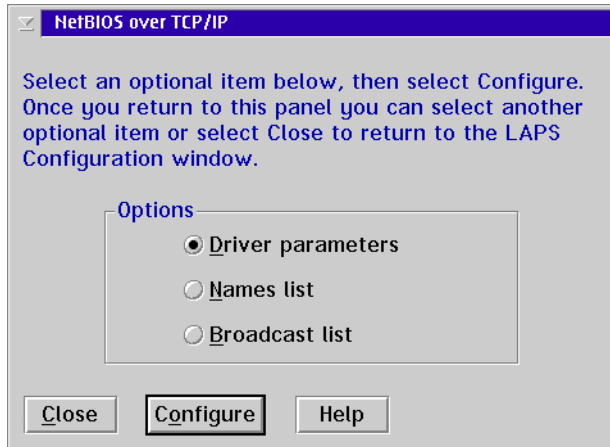


Figure 137. [OS/2 Warp] NetBIOS Over TCP/IP Configuration Options

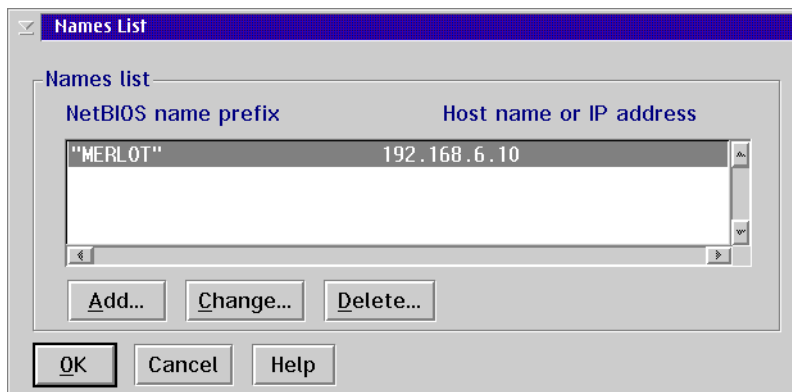


Figure 138. [OS/2 Warp] Modifying RFCNAMES.LST through MPTS

The advantage of using a names file is the reduction of name query broadcasts. Before the computer broadcasts to the network for NetBIOS name to IP address resolution, it searches this local file. The search is done by checking each names file entry up to the length of the entry. Due to the search technique used, the order of entries in the names file is important.

If an address in the file is given as a host name string, it is resolved to an IP address by looking it up in the local hosts file or by querying the Internet domain name server. The host name string must therefore be in the same form here as it is in the hosts file or on the domain name server.

For an OS/2 Warp Server environment, the requesters need a names file with computer name, IP address and domain name IP address of each server they contact. The servers need the computer name and IP address for each requester that contacts them. The computer name and domain names are found in the IBMLAN.INI files on these machines.

Broadcast File Routing Extension

This routing extension is implemented through the use of a broadcast file that contains the IP addresses of other subnets. A TCP/IP broadcast is limited to one subnet unless other subnet addresses are included in the broadcast file.

The broadcast file, named RFCBCST.LST is located in the \IBMCOM directory. It contains a list of host names (such as merlot), host IP addresses (such as 192.168.6.10) or subnet broadcast addresses (such as 192.168.6.255). The file is read once at startup and each valid address is added to the set of destination addresses for broadcast packets. Any host or network accessible by these addresses becomes part of the local NetBIOS broadcast domain; that is, they receive all name lookup and registration packets, as well as all broadcast and group NetBIOS datagrams. These addresses can be on other networks and accessed through the appropriate router or bridge by the IBM TCP/IP program. The remote node is treated as if it were a node on the local network.

Each line of the broadcast file is either an IP address in dotted decimal notation or a host name string that occurs in either the local hosts file used by the IBM TCP/IP program or on an Internet domain name server. The host name string must be in the same form here as it is in the hosts file or on the domain name server.

Again, the broadcast list may be edited manually, or through MPTS.

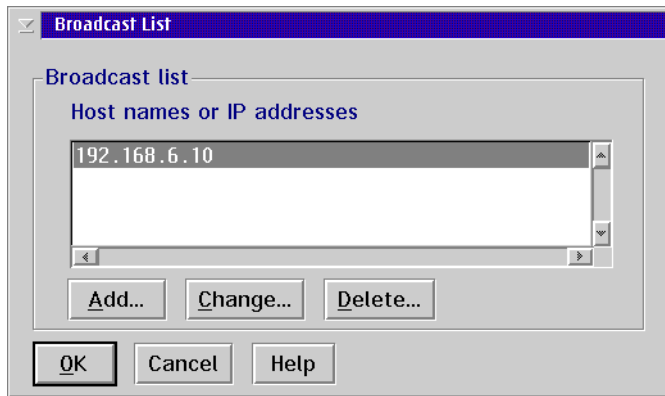


Figure 139. [OS/2 Warp] Modifying RFCBCST.LST through MPTS

For OS/2 Warp Server and OS/2 Warp File and Print Clients, if the server and requester are on different networks, the broadcast file must be updated. At the clients, the broadcast file must include the IP address or TCP/IP host name for each server that this requester accesses using TCPBEUI. At the servers, the broadcast file must include the directed broadcast address for each subnet containing LAN requesters.

The following must apply when using a broadcast file:

- IP addresses, subnet broadcast addresses and host name strings are allowed.
 - A maximum of 32 entries is allowed.
- Note:** You will find a higher number of entries supported in the upcoming OS/2 Warp Server for e-business, due 1Q/1999.
- Host name strings should match those in the hosts file and domain name server.
 - Routers should be checked to ensure that they forward directed broadcasts.

RFCADDR.EXE Program

When you make changes to the names/broadcast files while TCPBEUI is active, you can reinitialize TCPBEUI with the new file by using the `RFCADDR` command. The cache will be updated with the new names.

Domain Name Server (DNS) Routing Extension

This routing extension uses the TCP/IP Domain Name Server (DNS), to store NetBIOS names and IP address pairs, so that a TCPBEUI client can obtain the IP address of another workstation from it. This extension requires that the workstations already have an IP name and address registered on the DNS, the network administrator then enters the NetBIOS name and the IP address of every TCPBEUI workstation into the DNS.

To enable the use of the Domain Name Server routing extension, you need to set the DOMAINSCOPE parameter in the PROTOCOL.INI file equal to your TCP/IP domain name. For example:

```
DOMAINSCOPE=austin.cooking.net
```

If the names file search fails, the domain scope string, if specified, is appended to the encoded NetBIOS name. The new host name string is then used to resolve the NetBIOS name to an IP address by looking up the name in the local TCP/IP hosts file or by querying the DNS. If the DNS knows the name, it sends back the IP address to the NetBIOS over TCP/IP program.

The advantage of this implementation is that you have a central repository of all NetBIOS names. However, you have to update your DNS with NetBIOS names that are in an encoded format. An encoded format is necessary because NetBIOS names can be 16 bytes of any bit pattern, but the domain name server only accepts a limited character set. The TCPBEUI program encodes them into the 32-byte, reversible, half-ASCII format that is specified in RFCs 1001/1002.

You can use the MAPNAME.EXE utility to encode NetBIOS names. MAPNAME is supplied in the \APPLETS\MPTSAPLT.ZIP file on MPTS diskette 5.

MAPNAME Syntax

```
MAPNAME <input> /FDBLXX
```

Where:

<input> Either: filename.LST or filename.RFC NetBIOS name. Maximum of 16 characters for a NetBIOS name, and 32 characters for a RFC name

Name can be any char.+ \xDD in any order, surrounded by "[]".
D is a hex digit (\xDD = one char.).

F	Indicates input from file. (default is name)
D	Indicates the mapping direction. Use r to indicate NetBIOS to RFC. Use n to indicate RFC to NetBIOS.
B	Append blanks up to the 16 character. (default is a null character)
Lxx	Indicates the last character. (xx can be any hex value)
***	If "F" is used, then the name can be the combination of any character that can be put in the file by the editor.

Once the names have been encoded, you can then store them in the DNS (or DDNS) so that they point back to the original host name. Remember that NetBIOS names are registered multiple times (with a different suffix) when a computer starts. So, each instance of name+suffix must be encoded and registered on the DNS.

Case-Sensitiveness

The NetBIOS and DOMAIN names must be encoded in upper-case when using MAPNAME.

Here's an extract from the DNS database before the encoded NetBIOS names are added.

MERLOT	86400	IN	A	192.168.6.10
		IN	HINFO	Domain Cont.
;				
BRIE	86400	IN	A	192.168.7.12
		IN	HINFO	Workstation
;				

Figure 140. [Warp Server] Sample DNS Database

Using the Warp Server MERLOT as an example, its name is registered three times with the 0x00, 0x03 and 0x20 suffixes. In addition, being on the domain ARMONK, it also registers the domain name with a 0x00 suffix.

Using MAPNAME for each instance of name+suffix, we encode the names into the following:

MAPNAME MERLOT /rbl00 gives the encoding ENEFFCEMEPFECACACACACACACACAAA

MAPNAME MERLOT /rbl03 gives the encoding ENEFFCEMEPFECACACACACACACAAD
MAPNAME MERLOT /rbl20 gives the encoding ENEFFCEMEPFECACACACACACACACACA
MAPNAME ARMONK /rbl00 gives the encoding EBFCENEPEOELCACACACACACACACAAA

Workstations, on the other hand, would normally register their names three times. They would register the NetBIOS name with the 0x00 and the 0x03 suffixes. Usually, they would also register the domain name with the 0x00 suffix as well, but seeing as this encoded name is already registered on the DNS, it cannot be registered again.

Using MAPNAME for the workstation BRIE, we encode the following:

MAPNAME BRIE /rbl00 gives the encoding ECFCEJEFCACACACACACACACACACAAA
MAPNAME BRIE /rbl03 gives the encoding ECFCEJEFCACACACACACACACACACAAD

The encoded names can now be added to the DNS database and pointed back to the TCP/IP host name (Using CNAME), where the IP address is specified. An extract from the new DNS database now looks like this:

MERLOT	86400	IN	A	192.168.6.10
		IN	HINFO	Domain Cont.
;				
ENEFFCEMEPFECACACACACACACACAAA	86400	IN	CNAME	MERLOT
		IN	HINFO	0x00 suffix
;				
ENEFFCEMEPFECACACACACACACACAAD	86400	IN	CNAME	MERLOT
		IN	HINFO	0x03 suffix
;				
ENEFFCEMEPFECACACACACACACACACA	86400	IN	CNAME	MERLOT
		IN	HINFO	0x20 suffix
;				
EBFCENEPEOELCACACACACACACACAAA	86400	IN	CNAME	MERLOT
		IN	HINFO	Domain 0x00
;				
BRIE	86400	IN	A	192.168.7.12
		IN	HINFO	Workstation
;				
ECFCEJEFCACACACACACACACACAAA	86400	IN	CNAME	BRIE
		IN	HINFO	0x00 suffix
;				
ECFCEJEFCACACACACACACACACAAD	86400	IN	CNAME	BRIE
		IN	HINFO	0x03 suffix

Figure 141. [Warp Server] DNS Database with Encoded NetBIOS Names.

5.3 New Way of Using DNS with Non-RFC-Encoded Name

The previous section informed you how to create the RFC-encoded names for server names and domain names. It is a huge effort for TCP/IP administrators to set up all NetBIOS names as static entries in the DNS files. Basically, the NetBIOS Name Server, as defined in the RFC 1001/1002, is the ideal solution for applications written to the NetBIOS over TCP/IP interface.

However, on the horizon there is a new way of NetBIOS name resolution:

- Non-RFC-encoded names in DNS. The Node Status request frame was introduced as a communication vehicle.

This new technique is introduced through Microsoft's Windows NT Service Pack 3 and can be implemented on Windows NT Workstation clients as well as Windows NT Server. As far as OS/2 Warp Server is concerned, there is a fix (APAR IC20646) for the server, but OS/2 Warp clients do not yet support this new technique.

Figure 142 illustrates the setup and interaction of Windows NT Server and the Windows NT Workstation client.

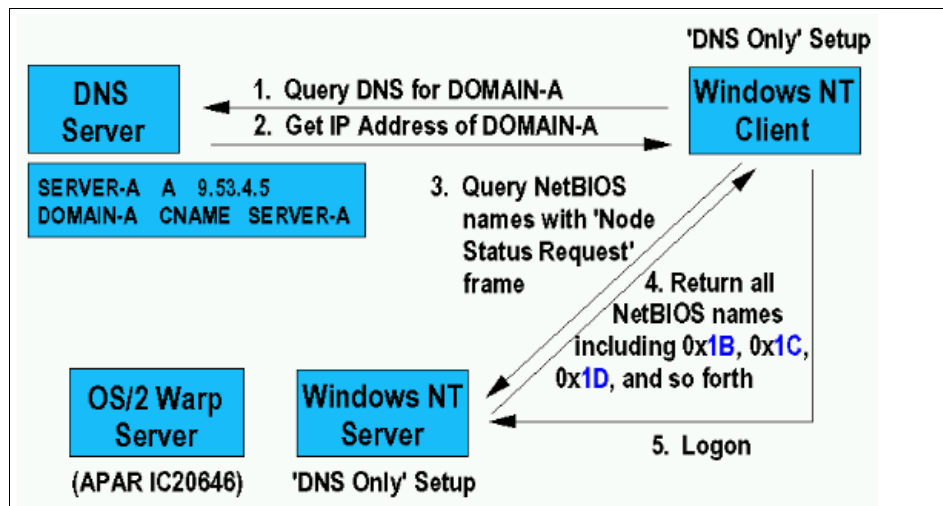


Figure 142. Using DNS Server with Node Status Request

In the DNS domain file, the Windows NT Server computername, also known as server name, is appended as a pure ASCII name with an A record that points to the server's static IP address. In addition, the domain name is

included with an alias, CNAME, to the server's name. The entries to the DNS domain file would look like the following two lines:

```
SERVER-A      IN      A      9.53.4.5
DOMAIN-A      IN      CNAME   SERVER-A.
```

The logon sequence would use the following steps:

1. The Windows NT Workstation client queries the DNS server for hostname DOMAIN-A.
2. The DNS server returns the information that DOMAIN-A is the alias to SERVER-A. The Windows NT Workstation client receives the IP address of the Windows NT Server.
3. Since the Windows NT Workstation client is not sure whether or not the target IP address is really a NetBIOS node which has certain NetBIOS names registered, it sends a RFC frame to the received IP address (which is the Windows NT Server). This frame is called Node Status request and passes along a resource record name of * (asterisk).
4. Windows NT Server responds to the Node Status request by sending a complete list of NetBIOS names registered in the node.
5. Now the Windows NT Workstation client selects the target IP address that has a correct server name and domain name, then it starts the logon sequence through a SMB session.

This technique might be called *DNS-Only* or *Use of non-RFC-encoded names* NetBIOS names resolution, but it actually is a combination of DNS name resolution and the Node Status Request frame defined in the RFC 1001/1002. Neither Windows NT Server nor Windows NT Workstation client use WINS in this case. The TCP/IP properties are set up as shown in Figure 143 on page 197 and Figure 144 on page 198.

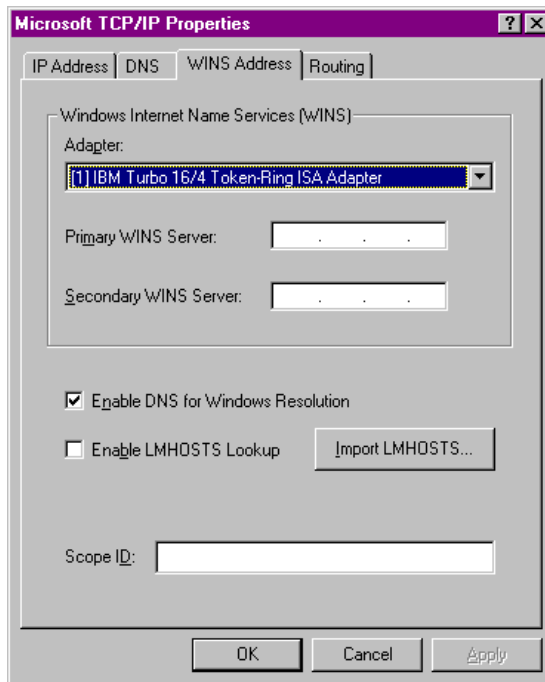


Figure 143. [Windows NT] WINS Address No Use of WINS but Use DNS

Windows NT will warn you when you don't provide the WINS server parameter at all. However, you can ignore the message and have the workstation configured to work with DNS only.

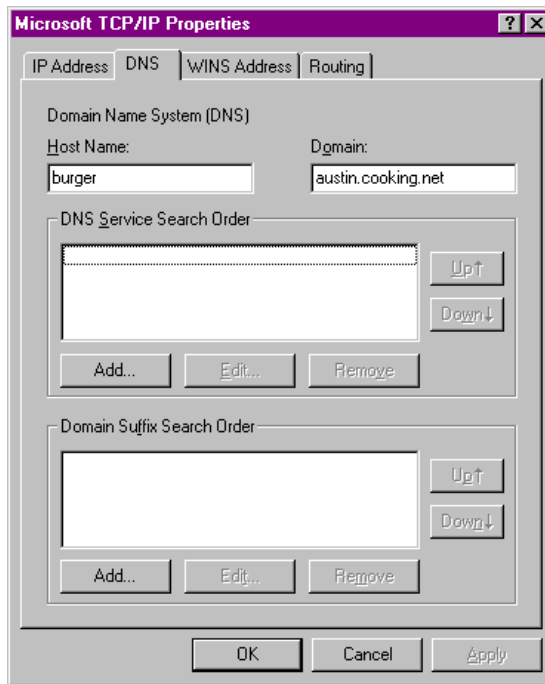


Figure 144. [Windows NT] TCP/IP Property - DNS Settings

Figure 145 on page 199 shows the two IP frames, a packet 49 sent from a Windows NT workstation carrying Node Status request frame with a questioned name *, and a packet 50 frame that was sent back from a Windows NT Server showing all registered NetBIOS names and status on the node.

The questioned name *(asterisk), followed by 15 bytes of null (0x00) characters, is actually translated into the RFC-encoded name that results in a character string of CKAAAAAAAAAAAAAAAAAAAAAAAAAAAA in the trace data.

```

Packet 49: IP,      9.3.1.138      -> 9.3.1.188
Source IP: 9.3.1.138      , Destination IP: 9.3.1.188
Version: 04,      IP header length: 05 (32 bit words)
Service type: 0: Precedence: 0, Delay: Norm, Throug: Norm, Reliab: Norm
PROTOCOL: [17] UDP
UDP,      [137] -> [137]
Source port: [137] ,      Destination port: [137]
TCPIP NETBIOS (name service)
Operation: [0] NAME QUERY REQUEST,      Flags: Unicast,
Transaction ID: 0803
Header Counts: QD: 1,      AN: 0, NS: 0, AR: 0
Request Name: *
Request type: NODE STATUS REQUEST

RAW PACKET LISTING:
0000 18 40 00 04 AC EC 35 D0      80 00 83 F4 6B 13 06 40      .@..-i5D.ôk..@
0010 40 31 40 10 AA AA 03 00      00 00 08 00 45 00 00 4E      @l@.ª.....E..N
0020 81 29 00 00 20 11 04 2B      09 03 01 8A 09 03 01 BC      ).. ..+.....¿
0030 00 89 00 89 00 3A A2 E0      08 03 00 00 00 01 00 00      ...:çà.....
0040 00 00 00 00 20 43 4B 41      41 41 41 41 41 41 41 41      .... CKAAAAAAAAAA
0050 41 41 41 41 41 41 41 41      41 41 41 41 41 41 41 41      AAAAAAAAAAAAAAAA
0060 41 41 41 41 41 00 00 21      00 01      AAAAA..!..
-----

Packet 50: IP,      9.3.1.188      -> 9.3.1.138
Source IP: 9.3.1.188      , Destination IP: 9.3.1.138
Version: 04,      IP header length: 05 (32 bit words)
Service type: 0: Precedence: 0, Delay: Norm, Throug: Norm, Reliab: Norm
PROTOCOL: [17] UDP
UDP,      [137] -> [137]
Source port: [137] ,      Destination port: [137]
TCPIP NETBIOS (name service)
Operation: [0] NAME QUERY RESPONSE,      Flags: Unicast,
Transaction ID: 0803
Header Counts: QD: 0,      AN: 1, NS: 0, AR: 0
Resource Record Name: *
Request type: NODE STATUS RESOURCE RECORD

Data
0000 54 53 4F 4E 54 30 30 20      20 20 20 20 20 20 00 44      TSONT00      .D
0010 00 49 54 53 4F 41 55 53      4E 54 20 20 20 20 20 20      .ITSOAUSNT
0020 00 C4 00 49 54 53 4F 4E      54 30 30 20 20 20 20 20      .Ä.ITSONT00
0030 20 20 20 44 00 49 54 53      4F 41 55 53 4E 54 20 20      D.ITSOAUSNT
0040 20 20 20 20 1C C4 00 49      54 53 4F 41 55 53 4E 54      .Ä.ITSOAUSNT
0050 20 20 20 20 20 20 1B 44      00 49 54 53 4F 41 55 53      .D.ITSOAUS
0060 4E 54 20 20 20 20 20 20      1E C4 00 49 54 53 4F 4E      NT      .Ä.ITSON
0070 54 30 30 20 20 20 20 20      20 20 03 44 00 49 54 53      T00      .D.ITS
0080 4F 41 55 53 4E 54 20 20      20 20 20 20 1D 44 00 01      O AUSNT      .D..
0090 02 5F 5F 4D 53 42 52 4F      57 53 45 5F 5F 02 01 C4      .__MSBROWSE__..Ä
00A0 00 49 4E 65 74 7E 53 65      72 76 69 63 65 73 20 20      .INet~Services
00B0 1C C4 00 49 53 7E 49 54      53 4F 4E 54 30 30 00 00      .Ä.IS~ITSONT00..
00C0 00 00 00 44 00 49 54 53      4F 4E 54 30 30 20 20 20      ...D.ITSONT00
00D0 20 20 20 20 01 44 00 41      44 4D 49 4E 49 53 54 52      .D.ADMINISTR
00E0 41 54 4F 52 20 20 03 44      00 00 04 AC EC 35 D0 00      ATOR      .D...-i5D.
00F0 00 00 00 00 00 00 00 00      00 00 00 00 00 00 00 00      .....
0100 00 00 00 00 00 00 00 00      00 00 00 00 00 00 00 00      .....
0110 00 00 00 00 00 00 00 2B      C3 80 B0 3E C3 80 14 2C      .....+Ã°>Ã.,
0120 C3 80 0F 00 01 00 00 00      00 00 00 00 00 00 00 00      Ã.....
0130 00 00 00 00 00 00 00 00      00 00 00 00 00 00 00 00      .....
0140 00 00 00 00 00 00 00 00      00      .....

```

Figure 145. [Windows NT] Packet Trace - Node Status Request

As shown in Figure 146, the `NETSTAT` command shows the formatted output of the Node Status Request frame.

```
C:\>NETSTAT -A 9.3.1.188
```

NetBIOS Remote Machine Name Table			
Name		Type	Status
ITSONT00	<00>	UNIQUE	Registered
ITSOAUSNT	<00>	GROUP	Registered
ITSONT00	<20>	UNIQUE	Registered
ITSOAUSNT	<1C>	GROUP	Registered
ITSOAUSNT	<1B>	UNIQUE	Registered
ITSOAUSNT	<1E>	GROUP	Registered
ITSONT00	<03>	UNIQUE	Registered
ITSOAUSNT	<1D>	UNIQUE	Registered
..__MSBROWSE__.	<01>	GROUP	Registered
INet~Services	<1C>	GROUP	Registered
IS~ITSONT00...	<00>	UNIQUE	Registered
ITSONT00	<01>	UNIQUE	Registered
ADMINISTRATOR	<03>	UNIQUE	Registered

MAC Address = 00-04-AC-EC-35-D0

Figure 146. [Windows NT] NBTSTAT Output

The CIFS (Common Internet File System) protocol as defined in IETF/RFC is supposed to extend the Internet to support the SMB file sharing protocol over the Internet. The idea of CIFS is the same as illustrated before:

- Non-RFC-encoded names in DNS
- Node Status request frame as communications vehicle

To make CIFS work properly, the name of `*SMBSERVER` followed by the blanks must not be used as a NetBIOS name by servers or workstations in a network. Using CIFS, clients do not have knowledge of server names or domain names. All SMB servers should respond to the `*SMBSERVER` name but not through a network broadcast. Clients are able to check whether or not a particular TCP/IP host is an SMB server.

The CIFS internet draft is located on the Internet at the following Web site:

<http://search.ietf.org/internet-drafts/draft-leach-cifs-v1-spec-01.txt>

It is still in informational status. There is a common belief that CIFS would lack security and that authentication logic is not strong enough on the public Internet. In addition to CIFS, remote administration and the network neighborhood browser protocol are discussed.

5.4 NetBIOS Name Servers (NBNS)

It would be much simpler if, instead of having to manually maintain configuration files on each computer, we could ask a NBNS to resolve the NetBIOS name to an IP address for us.

We will look at two NBNS. The first is WINS from Microsoft, the second is Shadow IPserver from Network TeleSystems.

5.4.1 Microsoft WINS

WINS is Microsoft's implementation of a NBNS. WINS only supports Microsoft's proprietary clients with its implementation of native NetBIOS and NetBIOS over TCP/IP

5.4.1.1 How WINS Works

Each Microsoft client needs to be configured with the IP address of a primary WINS server, and optionally with the IP address of a secondary WINS server.

Whenever a client (configured to use TCPBEUI and WINS) starts, it will attempt to register its NetBIOS name and IP address with the primary WINS server. The registration occurs when services or applications are started (For example, Workstation or Messenger), and is sent directly to the primary WINS server. If the name is not already registered to another client, the server responds with a message detailing the NetBIOS name that has been registered, and the Name Time to Live (*TTL*).

If after attempting three times to register its name with the primary server and failing, the client will attempt to register its name using the secondary server. If the secondary server also fails to respond, the client will revert to broadcasting in order to register its name.

The name registrations are made on a temporary basis, and the client is responsible for maintaining the lease of the registered name.

At one-eighths of the TTL, the client will attempt to refresh its name registration with the primary WINS server. If the client does not receive a response from the server, it will continue to attempt to refresh the registration

every two minutes until half the TTL has expired. At this point it will repeat the procedure, but this time using the secondary WINS server.

With WINS enabled, the client acts as an H-Node client for name registration. For resolution, it is H-Node with a few modifications. The sequence used by a WINS client for name resolution is:

- Check to see if it is the local machine name.
- Check the cache. (Any resolved name is placed in a cache for 10 minutes)
- Try to use the primary WINS server. (Use the secondary server if the primary does not answer after three attempts).
- Try a name query broadcast.
- Check the LMHOSTS file. (If the computer is configured to use LMHOSTS).
- Try the HOSTS file.
- Try the DNS.

5.4.1.2 System Requirements for WINS

Table 9 details the system requirements to implement a WINS server.

Table 9. WINS Hardware and Software Requirements

Type	Details
Hardware Requirements	
Processor	80486-33 (Pentium recommended)
Hard Drive	125 MB
Memory	16 MB (32 MB recommended)
Network Adapter	Any network adapter supported by Windows NT
Video	VGA or above
Software Requirements	
Windows NT	NT Server 4.0 with Service Pack 3
IP configuration	IP address Subnet mask Default router (These can be allocated by DHCP, but statically assigned parameters are recommended)

5.4.1.3 Installing WINS Server

(This procedure assumes that Windows NT has previously been installed, and that IP connectivity on the network is working)

1. From the Control Panel Network tool, select the **Services** tab.
2. Select the **Add** button.
3. Select **Windows Internet Name Service** from the list.
4. Select the **OK** button.
5. The WINS service will be installed.
6. Optionally add static entries for non-WINS clients.

Non-WINS clients can be statically added so that WINS clients can resolve their name without having to maintain a LMHOSTS file.

7. Optionally setup a WINS proxy agent.

A WINS proxy agent extends the functions of a WINS server to non-WINS clients. When the proxy agent 'sees' a name registration request, it forwards the request to a WINS server. The server verifies the name only, but does not register it. When the proxy agent 'sees' a query to resolve a name, it forwards the query to the WINS server. The server responds to the agent, who, in turn, returns the details to the client.

Important

The IP addresses manually configured for primary and secondary WINS servers take precedence over those supplied as DHCP options.

5.4.1.4 WINS Limitation

According to RFC 1001/1002, a NetBIOS name server should support all group names. WINS, however, only keeps a list of IP addresses for group names ending in 0x1C. Warp Server domains however, are registered with a 0x00 suffix and, as such, are not stored by a WINS server. Therefore, when an IBM client requests an IP address from WINS server, a broadcast address (an IP address of all ones) is returned. This makes it difficult for IBM clients to communicate across a routed network when using WINS as the NBNS.

5.4.2 Network TeleSystems Shadow IPserver

Shadow IPserver is a software system for managing name and address assignments and desktop configuration information within a TCP/IP network. IPserver includes a robust set of network services offered through standards-based protocol interactions, such as:

- DHCP (Dynamic Host Configuration Protocol) service
- DNS (Domain Name Server) service
- NBNS (NetBIOS Name Server) and NBDD (NetBIOS Datagram Distributor) services

The Shadow IPserver NBNS service is an integrated NetBIOS name server and datagram distributor. The IPserver NBNS service fully implements RFCs 1001 and 1002. The NBNS service provides name resolution services similar to those provided by Microsoft WINS.

Note: To use the datagram distributor, both the NBNS server and TCP/IP client stack must support datagram distribution, such as:

- OS/2 Warp Server with latest MPTS ServicePak
- OS/2 Warp 4
- DOS LAN Services (shipped with OS/2 Warp Server)

If you plan to use the Shadow IPserver Datagram Distributor functionality, you can use the Network TeleSystems TCP Pro protocol stack on the DLS client. The TCP Pro protocol stack fully supports Datagram Distribution. The Microsoft Windows 95 and Windows NT protocol stacks do not support datagram distribution.

The Shadow IPserver NBNS server supports both IBM and Microsoft NetBIOS networks; the same NBNS server can be used with OS/2 Warp, DOS/Windows workstations with DOS LAN Services installed, Windows 95, and Windows NT workstations.

The idea of the NBNS is that instead of broadcasting the NetBIOS names to the whole IP subnet, the names to be registered are sent to a NBNS server. The NBNS server then checks if the name is already in use, and if not, the name is registered. All the future queries to that name will be resolved with one single query to the NBNS.

To get the full advantage of the Shadow IPserver NetBIOS name server, install IBMs Neighborhood Browser Enabler for Warp Server software on the OS/2 Warp Server Domain Controller machine. This way the OS/2 Warp Server Domain Controller registers the Microsoft network dependent NetBIOS names and acts as the domain master browser machine. See 5.6 on page 222 for information on Neighborhood Browser Enabler.

5.4.2.1 System Requirements for Shadow IPserver

Table 10 details the system requirements to implement Shadow IPserver.

Table 10. Shadow Hardware and Software Requirements

Type	Details
Hardware Requirements	
Processor	80386 or above (80486-100 recommended)
Hard Drive	150 MB IDE/EIDE only (SCSI is not supported)
Memory	16 MB
Video	VGA
Network Adapter	One of the following network adapters 3Com 3C509 ISA Ethernet Eagle Technology's Novell NE2000plus3 ISA Ethernet IBM Auto 16/4 Token-Ring ISA Note: You can use the IBM Token Ring Turbo 16/4 if you use the LANAIID software provided by IBM and place the card in 16/4 automode.
Software Requirements	
Operating System	IBM or MS DOS 5.0 or later Note: The server software uses its own 32-bit operating system and does not run on top of DOS. However, you need to use DOS to install the server files and start the server program. Also be aware that only IBM PC DOS 2000 is Year 2000-compliant.
IP Configuration	IP address Subnet mask Default router

The configuration used for this test environment is:

- Shadow IPserver Version 3.021
- IBM PC-DOS 7.0 with Year 2000 fixes
 - IBM ValuePoint DX4-100
 - 64 MB memory
 - An IBM 16/4 ISA Token-Ring Adapter, configured manually:
 - 16 MBit/sec data rate
 - ROM address C800
 - IO port 0A20
 - RIPL disabled

5.4.2.2 Installing Shadow IPserver

The installation program (INSTALL.EXE) asks you for configuration information and makes entries in the NTS-SRVR.CFG file based on your input. To install IPserver, follow the steps below:

1. Make sure the following applies to the machine on which you will install Shadow IPserver:
 1. The whole hard disk must be configured as one primary partition.
 2. The hard disk must be formatted and installed with DOS.
 3. No memory management statements in the CONFIG.SYS and AUTOEXEC.BAT files. If there are any, like EMM386.EXE or RAMBOOST.EXE, remove those lines. HIMEM.SYS may be present, and is used by some of the Shadow IPserver utilities.
 4. Check the PC's time and date to verify that they are correct. If the time and date are not correct, set them using the DOS `TIME` and `DATE` commands.

Note: Shadow IPserver uses the PC's time and date to time stamp system events. If the server is halted and then restarted during operation, the server uses the time stamps on server Configuration and Save files to determine which files to load. This can affect the data that is loaded into the server's database.

2. Place the Shadow Installation diskette into the PC's diskette drive.
3. Start the installation program by issuing the following command:

```
<drive letter>:\INSTALL
```

where *<drive letter>* is the drive that contains the installation diskette.
4. When prompted, enter the required information such as the NBNS IP address, subnet mask and router address.
5. When prompted, select the appropriate network adapter and adapter settings.

Note: Your PC must contain one of the card types listed in Table 10 on page 205. If the PC does not contain one of the listed card types, you need to install one before using the server. Select the card type you plan to install and continue with the installation. See the *Shadow Server Reference Manual* for more information.

6. After the installation is finished, use a DOS editor, like E (PC-DOS) or EDIT (MS-DOS), to edit the C:\SHADOW\NTS-SRVR.CFG file, where SHADOW is the NBNS server installation directory. Check that the network adapter I/O port address and RAM address are correct.

There are over 60 keywords that can be used in the NTS-SRVR.CFG file to configure Shadow IPserver. Refer to Section 6.5.2.1, “The NTS-SRVR.CFG Configuration File” on page 260 for details.

Figure 147 shows the sample NTS-SRVR.CFG file used in our test environment:

```
#-----
# EVAL CONFIG #1
#-----
#                               Shadow IPserver #1 EVAL ONLY Configuration File
#-----
NOLOG                # Log files - Comment out to enable Logs
#HISTORY             # History file - Uncomment to enable History File
#AUTHORIZATION       # WEB Authorization - Uncomment to enable Authorization
#OUTAHERE            # Hit ESC key to exit IPserver
#NODIAGS             #
#RESOLVEPARENTS      #
#-----
#Server Configuration
#-----
NETSUBNETMASK        255.255.255.0
IPADDR               192.168.6.5
GATEWAYADDR          192.168.6.1
#BACKUPADDR          207.87.72.220
#DHCPBACKUPADDR      207.87.72.220
#DNSBACKUPADDR       207.87.72.220
#DNSCOSERVER         111.111.111.111,222.222.222.222
#DNSFORWARDER        111.111.111.111,222.222.222.222
#IPTIMETOLIVE        60      # IP time to live default = 60
#ARPTIMEOUT          120     # ARP timeout default = 120
TTL                  10800   #(235903197=max) time to live (in seconds) for
                        # dynamic entries - NBNS
#SYNCTTL             20     # Sync time to live - NBNS
#RELEASETTL          10     # Release time to live - NBNS
#NEGACACHETTL        360    # Negative response cache time to live - DNS
#-----
#Hardware parameters
#-----
NIC IBMTR            # Adapter type
# Supported Adapter Types:
#   NIC 3C509         # 3COM 3C509
#   NIC NE2000        # Eagle NE2000
#   NIC GpChiu        # UB Networks NIUpc/EOTP
#   NIC PCniuEX       # UB Networks pcNIU/ex 512K
#   NIC IBMTR         # IBM Auto 16/4 Token-Ring ISA
#
CONNECTOR TPI        # Twisted pair
# Supported Connector Types:
#   CONNECTOR TPI     # Twisted pair
#   CONNECTOR THINNET # Thin net
#   CONNECTOR THICKNET # Thick net
#   CONNECTOR STARLAN # Starlan
IO-PORT              0A20    #
WINDOW               0C800   # NIU memory window
```

Figure 147. [Shadow IPServer] NTS-SRVR.CFG Configuration File

The settings changed manually in the test environment were:

- The OUTAHERE parameter was remarked. This prevents Shadow from exiting if you accidentally hit the Esc key. To exit Shadow press **Ctrl-Right Shift-Esc**
- The TTL (Time To Live) setting was set to 10800 seconds (3 hours) to give optimal support for Windows 95 clients.

Note: If you do not have any Windows 95 clients in you network, you can set this value differently, for example, to100.

7. Check that the command `NTS-SRVR` was added to the `AUTOEXEC.BAT` file to autostart IPserver.
8. Optionally, modify the `NTS-SRVR.NBN` file to add entries to the NBNS database on the server at startup. Refer to 5.4.2.3 for the format of the `NTS-SRVR.NBN` file.
9. If you intend to use the server as a DHCP and DDNS server, modify the `NTS-SRVR.DHC` and `NTS-SRVR.DNS` files as required. Refer to Section 6.5, “DHCP and DDNS with Shadow IPserver” on page 258 for more information.

5.4.2.3 The NTS-SRVR.NBN File

The `NTS-SRVR.NBN` file is used by Shadow to pre-load NetBIOS names into the NBNS database at startup. The file can be created with any text editor.

Entries are added one per line, with the format:

`<name> <name_type><node_type> <ip_address> <entry_type>`

Note that there is no space between `<name_type>` and `<node_type>`.

Where:

`<name>` The NetBIOS name of the client. Names must be 16 characters long. Characters that cannot be entered can be represented by pairs of hexadecimal digits surrounded by square brackets.

To enter spaces in `<name>` use a hyphen character.

To enter a hyphen in `<name>`, use the `[20]` sequence.

`<name_type>` Either `U` for a unique name or `G` for a group name.

`<node_type>` Either `B` for Broadcast node, `P` for Point-to-Point node, `H` for Hybrid node or `M` for Mixed node.

<ip_address> The dotted decimal address of the client.

<entry_type> Either **S** for static or **D** for dynamic.

Figure 148 shows a sample NTS-SRVR.NBN file that contains an entry for the Shadow server, and entries for the Warp Server domain controller MERLOT on the domain ARMONK. MERLOT is also running the Neighborhood Browser Enabler hence the entries with the 1B, 1C, 1D and 1E suffixes and the MSBROWSE entry.

```
SHADOW UH 192.168.6.5 D
MERLOT-----[00] UH 192.168.6.10 S
MERLOT-----[03] UH 192.168.6.10 S
MERLOT UH 192.168.6.10 S
ARMONK-----[00] GH 192.168.6.10 S
ARMONK-----[1B] UH 192.168.6.10 S
ARMONK-----[1C] GH 192.168.6.10 S
ARMONK-----[1D] UH 192.168.6.10 S
ARMONK-----[1E] GH 192.168.6.10 S
[0102]__MSBROWSE__[0201] GH 192.168.6.10 S
```

Figure 148. [Shadow IPserver] Sample NTS-SRVR.NBN file

5.5 P-Node, H-Node and M-Node Clients

After installing a NBNS, all the servers and clients running NetBIOS over TCP/IP need to be configured to use the NBNS server. Notice that servers and clients normally register their NetBIOS names when the requester services are started.

Clients can have the IP address of the NBNS manually coded, or they can receive the address as options with their DHCP configuration.

5.5.1 Manually Configuring Clients to Use a NBNS

The following information demonstrates how to manually configure a client to make use of a NBNS

5.5.1.1 DOS, Windows 3.1 and Windows for Workgroups

To have DOS and Windows clients running DLS communicate with a NBNS (rather than using the NBUTIL utility), the \NET\PROTOCOL.INI file must be modified with the addition of statements to the [NTS\$NTSTS] (or [TCP\$PRO2]) sections.

Note

These statements cannot be added or configured through the DLS installation program.

Use a text editor such as E, EDIT or NOTEPAD to modify the PROTOCOL.INI to add the following options

```
NODETYPE=H-Node
NBNSAddr=[NBNS_ipaddress]
NBDGAddr=[NBDD_ipaddress]
```

where:

[NBNS_ipaddress] Is the IP Address of the NBNS

[NBDD_ipaddress] is the IP Address of the Datagram Distributer

Figure 149 shows an extract from a PROTOCOL.INI file, with the statements to support communications with a NBNS highlighted.

```
[NTS$NTST2]
DNSAddr=
GatewayAddr=
NetSubNetMask=
IPAddr=
TCPHeartBeats=Standard
DHCPClientID=BURGER
BootPFlag=DHCP
NODETYPE=H-Node
NBNSAddr=[192.168.6.5]
NBDGAddr=[192.168.6.5]
DriverName=ntsts$
VCs=16
VCReceiveLarge=6
VCSends=6
RcvWindow=2920
UseMemory=UMB
BINDINGS=IBM$GENIBMTOK
LANABASE=0
Token-Ring
```

Figure 149. [DOS LAN Services] PROTOCOL.INI Extract Showing NBNS Setup

5.5.1.2 Windows 95

In Windows 95 the NBNS server IP address has to be defined as a WINS server. The operating system takes care of the proper node-type selection. Set the WINS address the following way:

1. Go to the Control Panel and double-click the **Network** icon.
2. Select the TCP/IP protocol from the list and click the **Properties** button.
3. Select the **WINS Configuration** tab, and choose to Enable WINS resolution. See Figure 150.

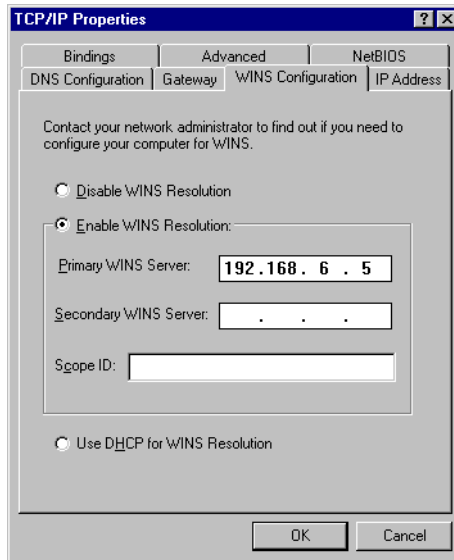


Figure 150. [Windows 95] Manually Configuring a NBNS

4. Enter the NBNS server IP address as the primary WINS server.
5. Optionally enter the IP address for a backup NBNS as the secondary WINS server.
6. Save the changes and reboot the machine to make the new settings effective.

5.5.1.3 Windows NT

In Windows NT the NBNS server IP address has to be defined as a WINS server. The operating system takes care of the proper node-type selection. Set the WINS address the following way:

1. Go to the Control Panel and double-click the **Network** icon.
2. Select the **Protocols** tab.

3. Select the TCP/IP protocol from the list and click the **Properties** button.
4. Select the **WINS Address** tab. See Figure 151

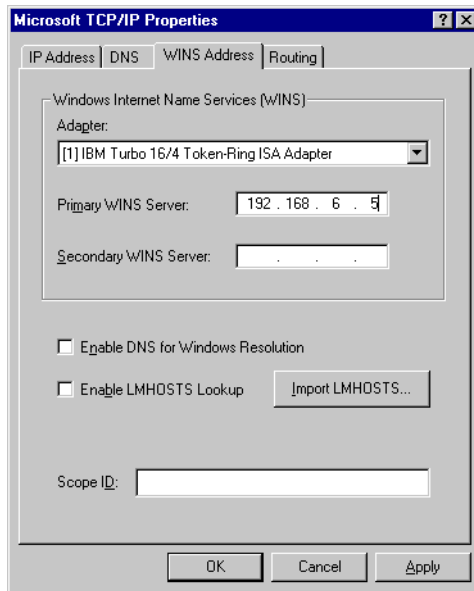


Figure 151. [Windows NT] Manually Configuring a NBNS

5. Enter the NBNS server IP address as the primary WINS server.
6. Optionally enter the IP address for a backup NBNS as the secondary WINS server.
7. Save the changes and reboot the machine to make the new settings effective.

5.5.1.4 OS/2 Warp

In order to have OS/2 Warp 4 clients communicate with a NetBIOS Name Server, you need to make changes to the default TCPBEUI settings. Using MPTS (Multi-Protocol and Transport Services) follow the steps below:

1. From System Setup (Warp Center pull-down menu) select **MPTS**.
2. Select **Configure** at the Multi-Protocol and Transport Services window.
3. Select the **radio** button for LAN adapters and protocols (which is usually set by default) and select **Configure**.
4. In the Adapter and Protocol window, make sure the current configuration list contains the IBM OS/2 NetBIOS over TCP/IP protocol. If IBM OS/2

NetBIOS is listed, the number for IBM OS/2 NetBIOS over TCP/IP must be set to 1. Double-click on **IBM OS/2 NetBIOS over TCP/IP**.

5. At the following NetBIOS over TCP/IP window select the **radio** button for Driver parameters and click on **Configure**.
6. At the Parameters for IBM OS/2 NetBIOS OVER TCP/IP window, make the following changes (Figure 152).
 - Set the Node Type parameter to H-node or P-node, depending on the need. In most situations it is recommended to use H-node. If P-node is used, registrations cannot be done without NBNS server. If H-node is used, the registrations are broadcasted if the NBNS server is not available.
 - Set the NetBIOS Name Server address parameter to the NBNS server IP address.
 - Set the NetBIOS Datagram Distribution server parameter to the NBNS sever IP address.
 - Optionally set the backup NBNS and NBDD server IP addresses.

Parameter	Value
Node Type	H-Node
Enable SLIP/PPP Interface	
NetBIOS Name Server address	192.168.6.5
Backup NetBIOS Name Server address	
NetBIOS Datagram Distributor address	192.168.6.5
Backup NetBIOS Datagram Distributor address	
NETBIOS trace level	0
Maximum sessions	225
Maximum commands	255
Maximum names	21
GDT selectors	15
Full buffer datagrams	NO
Query timeout	500
NETBIOS retries	2

Figure 152. [OS/2 Warp] Manually Configuring a NBNS

7. Select **OK** and **Close** as many times as necessary to exit MPTS and allow MPTS to update the CONFIG.SYS file if required.
8. To make protocol changes active, you need to reboot your workstation.

5.5.2 Dynamically Configuring Clients to Use a NBNS

A DHCP server is able to supply NBNS server information to DHCP clients using standard options. There are three options available to inform clients of the NBNS servers.

Option	Description
044 NBNS	The addresses of the primary and (optionally) the backup NBNS servers.
045 NBDD	The addresses of the primary and (optionally) the backup NetBIOS Datagram Distributor servers.
046 Node type	0x1 = B-Node, 0x2 = P-Node, 0x4 = M-Node, 0x8 = H-Node.

Note that pure WINS clients do not make use of a Datagram Distributor server, so option 045 may not be required in your environment.

5.5.2.1 DOS, Windows 3.11 and Windows for Workgroups

With DLS installed, all that is required is to ensure that PROTOCOL.INI does not contain the `NODETYPE`, `NBNSAddr` and `NBDGAddr` parameters.

The computer will then use options 44, 45 and 46 when it receives them.

5.5.2.2 Windows 95

In Windows 95 the computer needs to be configured to use a returned NBNS IP address

1. Go to the Control Panel and double-click the **Network** icon.
2. Select the TCP/IP protocol from the list and click the **Properties** button.
3. Select the **WINS Configuration** tab, and choose to Obtain an IP address from DHCP server.
4. Save the changes and reboot the machine to make the new settings effective.

5.5.2.3 Windows NT

With Windows NT, ensure that no WINS servers have been manually configured. Any manually configured NBNS servers have priority over NBNS IP addresses returned as DHCP options.

5.5.2.4 OS/2 Warp

At this point in time, OS/2 Warp Connect and OS/2 Warp 4 do not directly support options 44, 45 and 46. So, we looked into a work-around for this problem and came up with the following solution.

The solution uses two REXX scripts: `SETNBNS.COMD` and `GETNBNS.COMD`.

`SETNBNS` is run by the DHCP client when it receives the NBNS options. To do this, the `\MPTN\ETC\DHCPD.CFG` file requires the following entries:

```
option 44 exec "setnbns.cmd 44 %s"
option 45 exec "setnbns.cmd 45 %s"
option 46 exec "setnbns.cmd 46 %s"
```

When the options are received, `SETNBNS` creates a REXX queue, and places the parameters into the queue.

The following three figures display `SETNBNS.COMD`.

```
/* *****
 * SETNBNS.COMD
 * by Peter Degotardi
 * Revised April 21, 1998
 * SETNBNS
 * Return values: 0=successful, 1=unsuccessful
 * Accept DHCP options 44, 45 and 46 and put them in a queue where
 * GETNBNS.COMD can read them.
 * *****/
return_code = 1      /* Assume everything will fail */

/** add REXXUtil functions **/
call RxFuncAdd 'SysLoadFuncs', 'RexxUtil', 'SysLoadFuncs'
call SysLoadFuncs

/* get option number and data from the command line */
parse arg option_tag option_data

if ((option_tag = '') | (option_data = '')) then
do
return_code = 1                      /* wrong number of arguments */
exit return_code
end

/** strip leading and trailing spaces **/
option_tag = strip(option_tag)
option_data = strip(option_data)

Select
when (option_tag = 44) then          /* NBNS address */
do
if (check_IPAddr(option_data)) then
return_code = queue_it(option_tag||','||option_data)
end
```

Figure 153. `SETNBNS.COMD` (Part 1 of 3)

```

when (option_tag = 45) then
do
if (check_IPaddr(option_data)) then          /* NBDD address */
return_code = queue_it(option_tag||','||option_data)
end

when (option_tag = 46) then
do
call lineout '\setnbns.log', 'option_data='option_data''
select
when option_data = 8 then data = 'H-Node'
when option_data = 4 then data = 'M-Node'
when option_data = 2 then data = 'P-Node'
otherwise data = 'B-Node'
end
return_code = queue_it(option_tag||','||data)
end
otherwise
nop                                           /* unsupported_option */
end

exit return_code

/*****
* check_IPaddr
* Check a dotted decimal IP address for valid format.
*
* Return 0 if there are not 4 octets or the data is out of range or
* there are embedded blanks in the address, else return 1.
*****/

check_IPaddr: Procedure
parse arg addr
addr = strip(addr)
parse var addr octet.1 '.' octet.2 '.' octet.3 '.' octet.4

do pos = 1 to 4
if (verify(octet.pos, '0123456789', N) \= 0) then
return 0                                     /* not an unsigned number */
if ((octet.pos < 0) | (octet.pos > 255)) then
return 0
end
return 1

/*****
* hex2ascii_string (option_data)
*
* Take a hex string in the form 'hex"nn nn nn ... ', verify it's a valid
* hex string, and convert it to an ASCII character string.
*
* Returns the string if the data is valid, a null string if not.
*****/

hex2ascii_string: Procedure
parse arg 'hex' ' ' data ' '
if (\ DATATYPE(data, X)) then return ''
data = strip(data, 'B', '09'x)
return X2C(data)

```

Figure 154. SETNBNS.CMD (Part 2 of 3)

```

/*****
* queue_it
* Put the data in a queue where it can be read by GETNBNS.
* Tries to create the queue. If a different name is returned, the queue
* must already exist, so delete the 'wrong' queue, set the queue to the
* 'correct' one, and put the data there.
*****/

queue_it: Procedure
parse arg data

qname = 'NBNSINFO'                      /* The queue we're using */

new_queue = rxqueue("create", qname)
if new_queue \= qname then
    rc = rxqueue("delete", new_queue)
rc = rxqueue("set", qname)
queue data
return 0

```

Figure 155. SETNBNS.CMD (Part 3 of 3)

GETNBNS.CMD is the second script and is called from STARTUP.CMD with a GETNBNS /GO (We use the /GO parameter so that users don't 'accidentally' run the script) It reads the queue created by SETNBNS and will rewrite PROTOCOL.INI if new parameters have been received. The script will wait a maximum of 60 seconds for new parameters, and will give up if none are received.

If we have re-written PROTOCOL.INI, a re-boot will be required to use the new options. Rather than automatically re-booting the computer, a message is displayed informing the user to do so.

The following four figures display GETNBNS.CMD.

```

/*****
* GETNBNS.CMD
* by Peter Degotardi
* Revised April 21, 1998
* GETNBNS /GO
* Return values: 0=successful, 1=no TcpBEUI, 2=Can't read PROTOCOL.INI
*               -1=timeout waiting for TcpBEUI options
*               99=run without specifying the /GO parameter
*
* Read DHCP options 44, 45 and 46 from the queue created by SETNBNS.CMD
*****/

/* Mandatory flag so user's don't accidentally run this */
arg run_flag
if run_flag \= '/GO' then exit 99

return_code = 1          /* Assume everything will fail */
is_data = 1             /* There might be data */
are_changes = 0         /* Have changes been made to PROTOCOL.INI */
nbns_queue = 'NBNSINFO' /* The queue input queue holding NBNS data */
prot.0 = 0              /* Stem for holding PROTOCOL.INI in memory */

/**** add REXXUtil functions ****/
call RxFuncAdd 'SysLoadFuncs', 'RexxUtil', 'SysLoadFuncs'
call SysLoadFuncs

/* Where's PROTOCOL.INI? */
boot_drive = filespec('drive', value('comspec', , 'os2environment'))
prot_ini = boot_drive || '\IBMCOM\PROTOCOL.INI'

/* Does this machine have TcpBEUI ? */
if tcpbeui_chk() = 0 then
do
call delete_queue
exit 1
end

rc = rxqueue("set", nbns_queue)
if queued() = 0 then
do t = 60 to 10 by -10
is_data = 0          /* Nope, no data */
say 'Waiting 't' seconds for new TcpBEUI options.'
call sysssleep(10)
if queued() \= 0 then /* We've seen some data, so */
do /* get out of here */
is_data = 1
leave
end
end

if is_data then
do
if read_prot() = 0 then
do forever
if queued() = 0 then leave
parse pull item
parse value item with option, 'data'
data = strip(data)

```

Figure 156. GETNBNS.CMD (Part 1 of 4)


```

select
when option = 44 then
do
call set_nbns data
end
when option = 45 then
do
call set_nbdd data
end
when option = 46 then
do
call set_node data
end
otherwise
nop
end
call SysSleep(5)           /* Give time for more options to arrive */
end
else
exit 2                     /* Couldn't read PROTOCOL.INI */

end

else
do
say 'No new TcpBEUI parameters received.'
say 'Giving up.'
exit -1
end

if are_changes then
do
rc = write_prot()
say 'Changes have been made to TCPBEUI options in PROTOCOL.INI.'
say 'Please reboot to activate them.'
say ''
say 'Press Enter to continue...'
pull entkey
end
exit 0

/*****
* tcpbeui_chk
* Check to see if tcpbeui is configured on this machine. It's not worth
* continuing if it isn't
*****/
tcpbeui_chk:
call SysFileSearch prot_ini, '[TCPBEUI_NIF]', has_tcpbeui
return has_tcpbeui.0

/*****
* delete_queue
* Delete the queue created by SETNBNS
*****/
delete_queue:
rc = rxqueue("delete", nbns_queue)
return

```

Figure 157. GETNBNS.CMD (Part 2 of 4)

```

/*****
* read_prot
* Read PROTOCOL.INI into memory
*****/
read_prot:
rc = stream(prot_ini, 'c', 'open read')
if rc = 'READY:' then
do
p = prot.0
do while lines(prot_ini) \= 0
p = p + 1
prot.0 = p
prot.p = linein(prot_ini)
end
rc = stream(prot_ini, 'c', 'close')
return 0
end
else
return 1

/*****
* write_prot
* Write the changed PROTOCOL.INI to disk
*****/

write_prot:
prot_bak = prot_ini || '.OLD'
call SysFileDelete(prot_bak)          /* Not checking for ReadOnly file */
'@COPY 'prot_ini' 'prot_bak' >NUL'    /* Not checking for OK copy */
call SysFileDelete(prot_ini)          /* Not checking for OK delete */
rc = stream(prot_ini, 'c', 'open write')
if rc = 'READY:' then
do
do p = 1 to prot.0
call lineout prot_ini, prot.p
end
rc = stream(prot_ini, 'c', 'close')
say 'Writing 'prot.0' lines to 'prot_ini
return 0
end
else
return 1

/*****
* set_node
* Set the Node Type in PROTOCOL.INI if different.
*****/
set_node:
parse arg nodetype
do p = 1 to prot.0
if left(strip(translate(prot.p)), 8) = 'NODETYPE' then
do
parse value prot.p with .=' current_type
parse value current_type with '"current_type"'
if translate(current_type) \= translate(nodetype) then
do
are_changes = 1
prot.p = '    NODETYPE = "' || nodetype || '"'
end
end
end
end

```

Figure 158. GETNBNS.CMD (Part 3 of 4)

```

        say 'NODETYPE was 'current_type', replacing with 'nodetype'
    end
end
end
return
/*****
* set_nbns
* Set the NBNSADDR in PROTOCOL.INI if different.
*****/
set_nbns:
parse arg nbns_addr
do p = 1 to prot.0
    if left(strip(translate(prot.p)), 8) = 'NBNSADDR' then
        do
            parse value prot.p with .'= ' current_addr
            parse value current_addr with '" 'current_addr'"
            if current_addr \= nbns_addr then
                do
                    are_changes = 1
                    prot.p = '    NBNSADDR = " ' || nbns_addr || '" '
                    say 'NBNSADDR was 'current_addr', replacing with 'nbns_addr'
                end
            end
        end
    end
end
return

/*****
* set_nbdd
* Set the NBDDADDR in PROTOCOL.INI if different.
*****/
set_nbdd:
parse arg nbdd_addr
do p = 1 to prot.0
    if left(strip(translate(prot.p)), 8) = 'NBDDADDR' then
        do
            parse value prot.p with .'= ' current_addr
            parse value current_addr with '" 'current_addr'"
            if current_addr \= nbdd_addr then
                do
                    are_changes = 1
                    prot.p = '    NBDDADDR = " ' || nbdd_addr || '" '
                    say 'NBNSADDR was 'current_addr', replacing with 'nbdd_addr'
                end
            end
        end
    end
end
return

```

Figure 159. GETNBNS.CMD (Part 4 of 4)

Technical Caveats

Be aware that the above scripts are not perfect, but are aimed to give you an idea as to how to process these options. We therefore draw your attention to some of the issues.

- SETNBNS.CMD will run every time a lease is renewed. As such, the REXX queue NBNSINFO will be re-created and data queued. With a short

lease time on a machine that is rarely restarted, you could end up with a queue containing a large quantity of data.

- Options 44 and 45 can contain multiple IP addresses, `SETNBNS` and `GETNBNS` cannot cope with this.
- Normally, the exec entries in `DHCPD.CFG` for a REXX script would be:

```
option 44 exec "$setnbns.cmd 44 %s"
```

The '\$' parameter is to force a 30 second wait so that REXX can initialize. We found that by including the '\$' parameter, a 30 second wait was introduced between each instance of `SETNBNS` running for each option, that is, a total of 90 seconds was required if all three options were processed. With this delay, `GETNBNS` only ever saw the first option queued. Without the '\$' `SETNBNS` and `GETNBNS` worked correctly.

- Error handling could be greatly improved. As you can see from the comments in `GETNBNS`, the backup/deletion/re-writing of `PROTOCOL.INI` could be improved.
- `GETNBNS` can be run from the Startup folder, but be aware that the user may not see the message that a restart is required.
- If the user chooses not to reboot, they may not be able to log onto the LAN.

5.6 IBMs Neighborhood Browser Enabler for Warp Server

IBM's Neighborhood Browser Enabler for OS/2 Warp Server, is an add-on for Warp Server that enables greater inter-operability between Microsoft clients and Warp Server.

5.6.1 Overview

The IBM Neighborhood Browser Enabler for OS/2 Warp Server (referred to Neighborhood Browser Enabler from now on) enables your OS/2 Warp Server to function as a master browser for Windows, Windows 95, and Windows NT clients. If you are using Windows 95 or Windows NT clients in an OS/2 Warp Server domain, and not using the Neighborhood Browser Enabler, you have to set one master browser per segment to be able to browse OS/2 Warp Server domain resources from Windows 95 and Windows NT. The master browser workstation has to be up and running all the time; it would be best to run this function on a server machine.

We recommend installing the Neighborhood Browser Enabler on the domain controller machine. Install it on any OS/2 Warp Server machine one master

browser at a time. Installing Neighborhood Browser Enabler on more than one server reduces the network traffic between the master browser and the clients. All the other servers running the Neighborhood Browser Enabler service are backup browsers. For performance considerations, it is recommended to install Neighborhood Browser Enablers for every 20 servers in the domain. Depending on the installation, this figure may vary.

5.6.2 Installing Neighborhood Browser Enabler

To install the Neighborhood Browser Enabler, follow the steps below:

1. Download the Neighborhood Browser Enabler as a .ZIP file from IBM's Software Choice at:

<http://service.software.ibm.com/asd-bin/doc/index.htm>

(This is a free download.)

2. From an OS/2 Window on your Warp Server, create a temporary directory, and copy the downloaded .ZIP file into it.
3. Unzip the package to the directory by using `PKUNZIP2`.

```
PKUNZIP2 BRINST.ZIP
```

4. Read the README.1ST file.
5. If the Server is running, stop it by entering the command:

```
NET STOP SERVER
```

6. Enter the command:

```
BRINST
```

This will install new files into their appropriate directories:

```
\IBMLAN\SERVICES\BROWSER.EXE
\IBMLAN\NETLIB\BROWSE32.DLL
\IBMLAN\NETLIB\BROWSE16.DLL
```

Depending on service level, selected base LAN Server modules are backed up, then newer versions are installed:

```
\IBMLAN\SERVICES\NETSERVR.EXE
\IBMLAN\SERVICES\NETSVINI.EXE
\IBMLAN\NETPROG\NETWKSTA.200
\IBM386FS\HPFS386.IFS
\MUGLIB\DLL\NETAPI32.DLL
```

Note: The file `\MUGLIB\DLL\NETAPI32.DLL` is replaced only on OS/2 Warp Server SMP IP08500.

7. A reboot may be necessary depending on LAN Server's service level.

The Neighborhood Browser Enabler installation program copies the browser files to the appropriate directories, after backing up the existing IBMLAN.INI file to IBMLAN.IBR, backing up some LAN server components and finally modifying the IBMLAN.INI file so that the browser service starts automatically when the server service is started.

The updated IBMLAN.INI file contains a new section:

```
[browser]
maintainserverlist = auto
```

It also adds a new service called BROWSER in the SRVSERVICES line, and it extends the path to new executable in [services] section.

5.6.3 Starting and Stopping the Neighborhood Browser Enabler

You can start the Neighborhood Browser Enabler service at any time by issuing the command:

```
NET START BROWSER
```

or stop the service by using the command:

```
NET STOP BROWSER
```

When Neighborhood Browser Enabler is installed on the OS/2 Warp Server domain controller machine, the domain controller also recognizes and answers to NetBIOS name types used by Microsoft networks but not by IBM networks. These types include domain controller, domain master browser, master browser, and browser service elections. In cases where Microsoft Windows clients are attached to the network, we recommend installing the Neighborhood Browser Enabler and using the appropriate IBM Networks Client for Windows.

5.7 Related Publications

- *Inside NetBIOS 2nd Edition*. Architecture Technology Corporation 1988. ISBN 0-939405-008. Haugdahl.
- *Inside OS/2 Warp Server, Volume 1: Exploring the Core Components*. IBM ITSO 1996 SG24-4786. Shimizu, Cepeda, Macedo, Murhammer, Naick, Odier, Pauli, Rykaert, Taylor, Testini, Vernon and Zimmerman.
- *Internetworking with Microsoft TCP/IP in Microsoft Windows NT 4.0*. Microsoft 1997 0871B

- *Network Clients for OS/2 Warp Server: OS/2 Warp 4, Windows 95/NT and Apple Macintosh*. IBM ITSO 1997 SG24-2009. Zimmerman, Moroian, Nummi, Persell and Wiberg.
- *OS/2 Warp Server, Windows NT and Netware: A Network Operating System Study*. IBM ITSO 1996 SG24-4786. Zimmerman, Berger, Chhana, Hinner, Pauli, Ronzoni, Scalmani and Wollert.
- *TCP/IP Implementation in an OS/2 Warp Environment*. IBM ITSO 1996 SG24-4730. Murphy, Grode, Lee and Wichmann.
- *TCP/IP Tutorial and Technical Overview (Fifth Edition)*. IBM ITSO 1995 GG24-3376. Murphy, Enders and Hayes
- *Understanding Performance Tuning Theory for IBM OS/2 LAN Server*. IBM ITSO 1994 GG24-4430. Shimizu, Gehrig, Takamura and Tambirasa.

Chapter 6. Growing Your Network

Planning and installing a simple DHCP and DDNS network is not a difficult task. However, when we need more DHCP servers or more (D)DNS servers, there are many complex design issues. We discuss multiple subnets, multiple DHCP servers, address pool management, DHCP relay agents, combination of a static DNS and DDNS, a mix of NT DNS and OS/2 Warp Server DDNS and AIX DDNS. If your network is growing, you'll want to read this chapter.

6.1 Multiple Subnets

There are several reasons why you want to divide your TCP/IP network with the subnets. Assuming a small company, ABC Corp., had been using one floor of the office building downtown, and now the company expands one more floor. If the newly added floor is next to the original floor, they might just extend the LAN, so that a bridge or repeater is enough to extend the TCP/IP network. However, if the LAN utilization is already high, the option is to use a router to physically divide two LANs. Thus you will end up with two subnets, and the router will connect two TCP/IP subnets on the IP level. In a very large building, each floor might be treated as a separate subnet, assuming each floor can have enough number of IP workstations, such as 255 or 512. There could be even more than one subnet, and the expensive router hardware supports multiple subnets in one box.

6.1.1 Static and Dynamic Routing Between Subnets

Routing is the process of choosing a network path on which to transfer packets. A *router* is a device that is (usually) connected to two or more networks and is responsible for transferring data from one network to another.

In order to perform routing, a router has to maintain a map of what subnets are on the network and where it should send data so that it can reach its destination. There are two ways of maintaining this list.

- You, as a network administrator, could manually configure the routing table on each router. This is said to be *static* routing. Static routing is easy to administer if your network is fairly small, but can become complex as your network continues to grow.
- The routers could "talk" amongst themselves to determine what subnets are on the network and how to get to them. Two common protocols used by routers are Routing Information Protocol (RIP) and Open Shortest Path First (OSPF).

Both OS/2 Warp and Windows NT include the functionality to act as routers. They can be configured as static or dynamic routers (or a combination of both).

Note

OS/2 Warp and Windows NT only support RIP. If you have routers that use OSPF (or other routing protocols), OS/2 Warp and Windows NT will not be able to exchange routing information with the other routers.

In showing how to configure OS/2 Warp and Windows NT as routers, we'll assume the sample network shown in Figure 160.

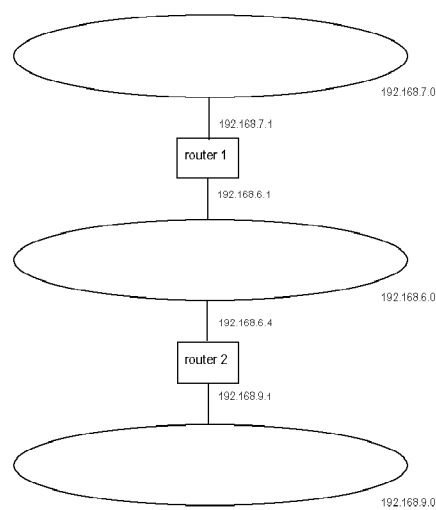


Figure 160. Sample Network for Routing

The routing tables required for this sample network are:

Table 11. Routing Tables Required for Sample Network

Router 1 Routing Table		Router 2 Routing Table	
Data for Network	Send to	Data for Network	Send to
192.168.7.0	192.168.7.1	192.168.7.0	192.168.6.1
192.168.6.0	192.168.6.1	192.168.6.0	192.168.6.4
192.168.9.0	192.168.6.4	192.168.9.0	192.168.9.1

6.1.2 OS/2 Warp as a Router

To enable OS/2 Warp 4 with TCP/IP Version 4.1 to act as a router, it should be connected to two or more networks and have statically assigned IP addresses for each network adapter.

6.1.2.1 Enabling Routing

To enable routing under OS/2 Warp 4, do the following:

- Using the TCP/IP configuration panels, switch to the **Routing** tab, and check the **IP Forwarding** checkbox, as shown in Figure 161.

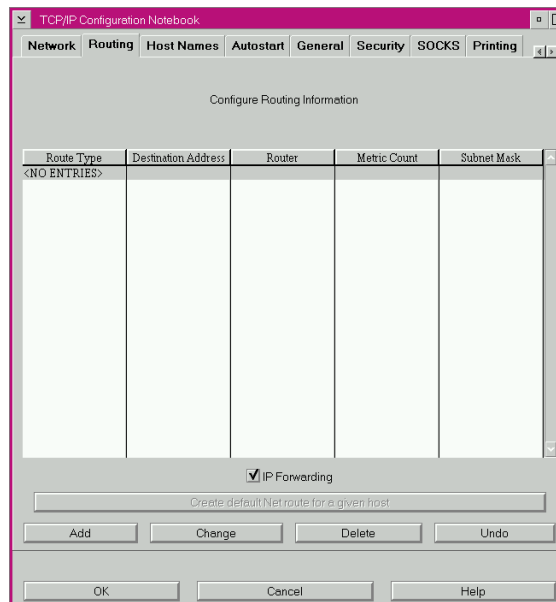


Figure 161. [OS/2 Warp] Enabling IP Forwarding

- Click on **OK**, and keep exiting until you have exited from the TCP/IP configuration program.

If you examine the `\MPTN\BIN\SETUP.CMD` file, you will see that the line

```
IPGATE OFF
```

has been changed to

```
IPGATE ON
```

- Shut down and restart.

6.1.2.2 Static Routing

To configure static routes under OS/2 Warp, do the following:

- Using the TCP/IP configuration panels, switch to the **Routing** tab, and click on the **Add** button. The Route Entry panel is displayed.

As shown in Figure 162, we are configuring Router 1 so that it knows where to send packets destined for the 192.168.9.0 subnet.

Route Entry

☒ After current list selection
☐ Before current list selection

Route Type
☒ net ☐ host ☐ default

Destination IP Address: 192.168.9.0
Router Address: 192.168.6.4
Metric Count: 1
Subnet Mask: 255.255.255.0

OK Cancel Help

Figure 162. [OS/2 Warp] Adding a Route

- Click on **OK**, and keep exiting until you have exited from the TCP/IP configuration program.

If you now examine \MPTN\BIN\SETUP.CMD file, you will see that any routes that you have added from the Route Entry panel have been inserted as `route` commands. For example, the entry for router 2 will be:

```
route add -net 191.168.7.0 192.168.6.1 netmask 255.255.255.0 -hopcount 1
```

- Shut down and restart.

ROUTE Syntax

```
ROUTE [-nqv] [add | delete] [-net | -host] dest gateway [-netmask mask]  
        [-nqv] [change] [-net | -host] dest [gateway] [-lock | -lockrest]  
        [-rtt | -rttvar | -sendpipe | -recvpipe | -mtu | -hopcount | -expire |  
        -ssthresh] value  
        [-nqv] [get] [-net | -host] dest  
        [-nqv] [monitor]  
        [-nqv] [flush]  
        [-nqv] [add | delete | change | get] [default] gateway
```

Where:

<code>-n</code>	Bypass translating IP addresses to symbolic hostnames.
<code>-q</code>	Suppress all output.
<code>-v</code>	Verbose.
<code>add</code>	To add a route.
<code>delete</code>	To delete a route.
<code>change</code>	To change aspects of a route.
<code>get</code>	To look up the route for a destination.
<code>monitor</code>	To continuously report changes to routes.
<code>flush</code>	To remove all routes.
<code>-net</code>	The following <i>dest</i> is a network address.
<code>-host</code>	The following <i>dest</i> is a hostname or address (default).
<code>-netmask</code>	The following is the mask of the route.
<i>dest</i>	The IP address or hostname of the destination.
<i>gateway</i>	The IP address or hostname of the next-hop router.
<i>mask</i>	The subnet mask.
<i>value</i>	A value.
<code>default</code>	0.0.0.0, for all destinations not defined by any other routes.

6.1.2.3 Dynamic Routing

For Dynamic routing, TCP/IP makes use of the `ROUTED` command. `Routed` enables RIP so that the computer can interact with other routers.

- Using the TCP/IP configuration panels, switch to the **Autostart** tab.
- From the Autostart Services window, select **routed**, as shown in Figure 163.

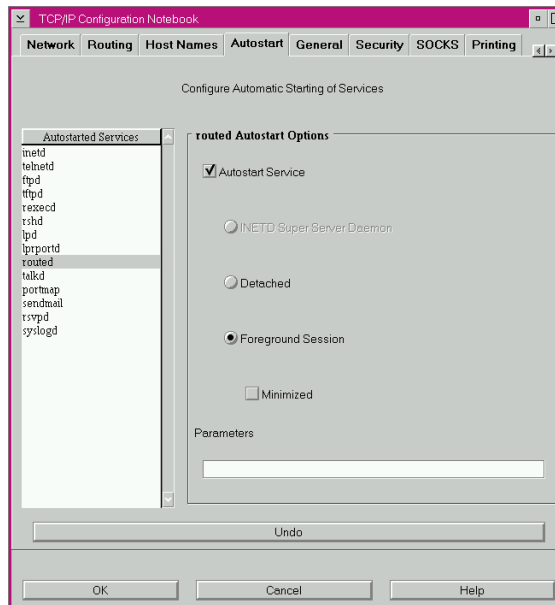


Figure 163. [OS/2 Warp] Enabling ROUTED

- Select the **Autostart Service** checkbox.
- Select either the **Detached** or **Foreground Session** radio button.
- If you selected Foreground Session, optionally select the **Minimized** checkbox.
- Optionally, set startup parameters. Refer to the online documentation for parameters that may be used with routed.
- Click on **OK**, and keep exiting until you have exited from the TCP/IP configuration program. If you now examine the `\TCP\BIN\TCPSTART.COM` file, you will see that a `start routed` command has been added.
- Shut down and restart.

6.1.3 Windows NT as a Router

As with OS/2 Warp, Windows NT should be connected to two or more networks with statically assigned IP addresses for each adapter before it can act as a router.

6.1.3.1 Enabling Routing

From the Control Panel, open the **Network** icon; then open the **Properties** for the TCP/IP protocol.

Select the **Routing** tab, and check the **Enable IP Forwarding** checkbox as shown in Figure 164.

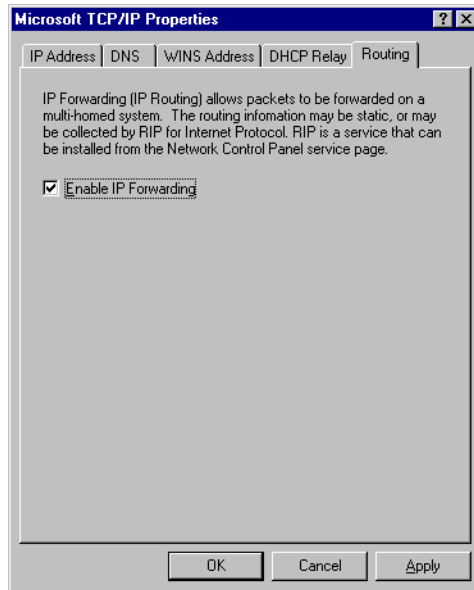


Figure 164. [Windows NT] Enabling IP Forwarding

Click on **OK** and exit from the Network configuration.

6.1.3.2 Static Routing

To add static routes to the routing table, you use the `route` command.

- For example, to permanently add the route for the 192.168.7.0 subnet to router 2, issue the command

```
route -p add 192.168.7.0 mask 255.255.255.0 192.168.6.1
```

ROUTE Syntax

```
ROUTE [-p] add network mask mask gateway metric metric  
          delete network gateway  
          change network gateway  
          print  
          -f
```

Where:

-p	Makes persistent changes. That is, changes are stored to the Registry.
add	Adds a route.
delete	Deletes a route
change	Changes a route
<i>network</i>	The destination subnet.
<i>mask</i>	The subnet mask.
<i>gateway</i>	Where to send packets destined for the subnet <i>network</i> .
<i>metric</i>	The cost of using this route.
-f	Flush the routing table.

6.1.3.3 Dynamic Routing

To perform dynamic routing, the RIP for Internet Protocol Service must be installed. This service comes as part of Windows NT Server.

- Open the **Network** icon in the Control Panel, and on the Services tab, select **Add**. A list of available services will be displayed.
- Scroll down the list, and select **RIP for Internet Protocol** (as shown in Figure 165 on page 235); then click on **OK**.

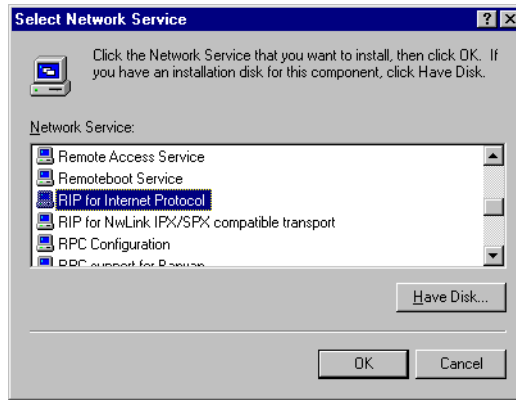


Figure 165. [Windows NT] Installing the RIP for Internet Protocol Service

- Keep exiting until you have exited from the network configuration.
- Shut down and restart.

6.2 Crossing Routers

Once your TCP/IP network begins to grow, you will for various reasons split the network into multiple subnets that are interconnected through routers.

Routers generally filter out all broadcast traffic between subnets. With this in mind, you should remember that a workstation that is being configured through DHCP is dependent on the DHCPDISCOVER and subsequent broadcasts being seen by a DHCP server.

So how do you make your DHCP clients work on one subnet when your DHCP server is on another subnet and your broadcast traffic is being filtered?

The simplest option is to configure your router to pass on all broadcast traffic. This will work, but will introduce further problems as your network continues to grow and is not recommended.

A better method would be to have some type of device on the subnet remote from the DHCP server listening for DHCP broadcasts, and when it sees one, it forwards it directly to the DHCP server.

RFC1542 Clarifications and Extensions for the Bootstrap Protocol, documents the methodology for implementing such a device. Although it was originally written with BOOTP in mind, it is valid for DHCP. A device that implements this type of forwarding is known as a DHCP Relay Agent (or a

BOOTP Relay Agent). Figure 166 shows the initial setup sequence of a DHCP client when the DHCP server is remote, but a Relay Agent is available.

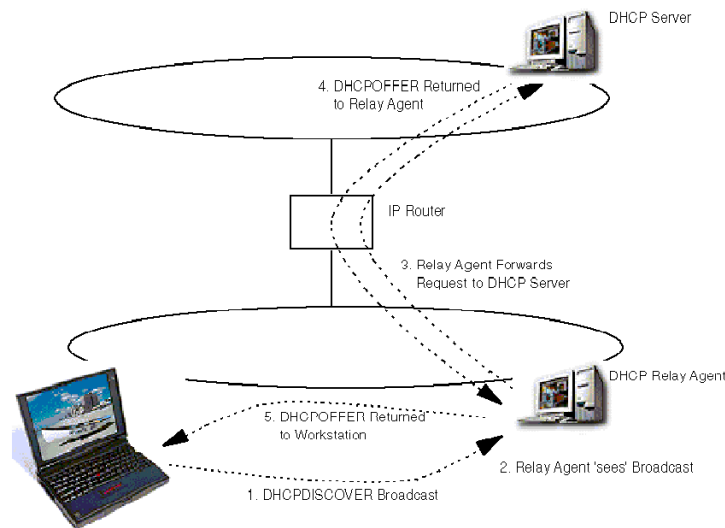


Figure 166. DHCP Initialization through a DHCP Relay Agent

You may have RFC 1542-compliant routers; if so, the Relay Agent can be run internally on the router. If your routers are not compliant, or if you do not want to add any additional processing load to the routers, there are software solutions available in Windows NT and OS/2 Warp to make standard computers act as DHCP Relay Agents.

Important

Before you consider running a DHCP Relay Agent on a machine that is itself a DHCP client, remember that it must be able to contact a DHCP server to get an IP address. It won't be able to do this because the Relay Agent is not running. Therefore, the Relay Agent should be run on a machine with a statically assigned IP address.

6.2.1 Windows NT as a DHCP Relay Agent

The DHCP Relay Agent is bundled with Windows NT Server, but is not installed by default.

Open the **Network** icon in the Control Panel, and on the Services tab, select **Add**. A list of available services will be displayed with the DHCP Relay Agent at the top of the list. Select it, and click on **OK**.

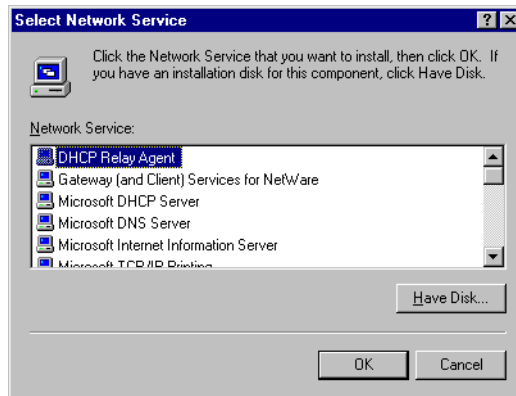


Figure 167. [Windows NT] List of Available Network Services

The Relay Agent will be installed and added to the list of installed services.

Click on **Close**. An error panel will be displayed stating that the DHCP Relay Agent cannot be used until a DHCP server IP address is configured. Click on **Yes** to add a DHCP server.

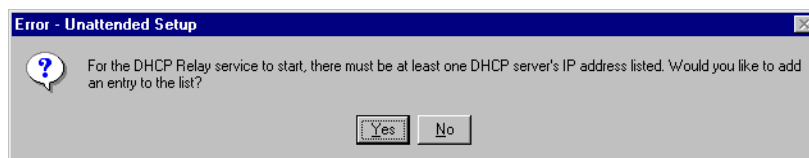


Figure 168. [Windows NT] Error Panel Shown after Relay Agent Install

The TCP/IP Properties panel is displayed. Select the **DHCP Relay** tab.

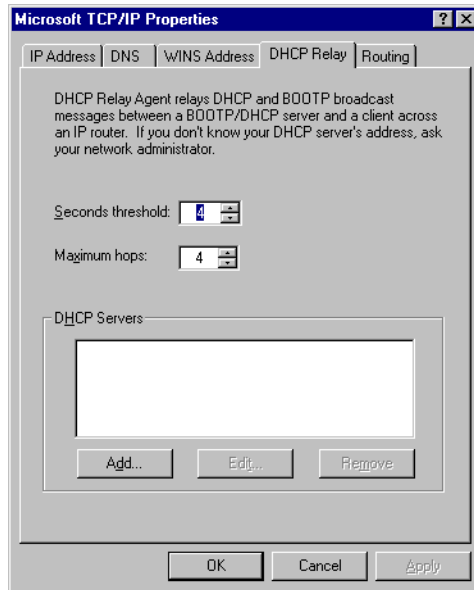


Figure 169. [Windows NT] DHCP Relay Tab

Click on the **Add** button. A panel will be displayed that will allow you to enter an address for a DHCP server. You can add multiple DHCP servers.

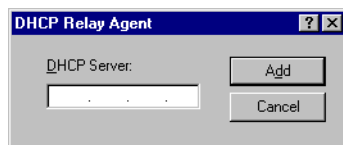


Figure 170. [Windows NT] Adding a DHCP Server

When you have finished adding the DHCP server addresses, click on the **OK** button to exit. As usual, you will have to reboot.

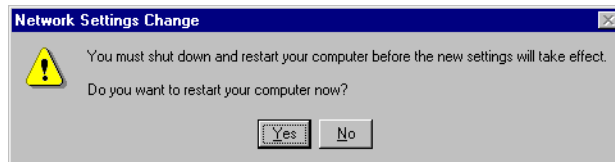


Figure 171. [Windows NT] Reboot Prompt

6.2.2 OS/2 Warp as a DHCP Relay Agent

Starting with TCP/IP Version 4.1 for OS/2 Warp, IBM has bundled a DHCP Relay Agent. Although information on how to set up the DHCP Relay Agent is not provided in the online documentation, you can obtain setup instructions as APAR IC20987 from the IBM Support Center or on the Web (by searching the APAR database) at <http://ps.software.ibm.com>. For your convenience, we have also provided those setup instructions in this section.

The Relay Agent consists of one executable, \TCPIP\BIN\DHCPD.EXE, and a required configuration file. By default, the configuration file \MPTN\ETC\DHCPD.CFG is used. A sample configuration file may be found in \TCPIP\SAMPLES\ETC as shown in Figure 172.

```
numLogFiles      4
logFileSize      100
logFileName      dhcprd.log
logItem          SYSERR
logItem          OBJERR
logItem          PROTERR
logItem          WARNING
logItem          EVENT
logItem          ACTION
logItem          INFO
logItem          ACNTING
logItem          TRACE

server 9.67.116.156
#
```

Figure 172. [OS/2 Warp] Sample DHCPD.CFG File

The configuration file must be manually edited using a text editor such as E or TEDIT. The options that can be contained within the file are as follows:

#	Comment
<i>numLogFiles</i>	The number of log files desired.
<i>logFileSize</i>	The size of log files in K bytes.
<i>logFileName</i>	The name of the most recent log file.
<i>logItem</i>	One item to be logged. The loggable items are:
SYSERR	Log system errors.
OBJERR	Log object errors between objects in the process.

PROTERR	Log protocol errors between client and server.
WARNING	Log warnings deserving attention from the user.
EVENT	Log events that occur to the process.
ACTION	Log actions taken by the process.
INFO	Log information that might be useful.
ACNTING	Log who was served when.
TRACE	Log code flow for debugging.
<i>server</i>	IP address of a single DHCP server. The server option may be specified multiple times so that DHCP requests are forwarded to multiple servers.

DHCPD Usage

```
DHCPD [-?] | [-b] | [[ -v | -q] [ConfigFile] ]
```

Where:

-?	Displays help message.
-b	Displays the program banner.
-q	Executes in quiet mode.
-v	Executes in verbose mode.
<i>ConfigFile</i>	Relay Agent configuration file. This file defaults to the following rules:
Name	dhcprd.cfg
Location	Searches the current working directory. If not there, the %ETC% directory (C:\MPTN\ETC) is searched.

Once the configuration file has been created, the Relay Agent can be started in several ways:

- The `DHCPD` command may be added to \STARTUP.CMD.
- The `DHCPD` command may be added to \TCP\BIN\TCPSTART.CMD.
- An icon can be added to the Startup Folder.

6.2.3 DHCP Relay Considerations

6.2.3.1 The 'Broadcast Flag'

DHCP includes a method whereby devices unable to receive a packet with a specific IP address can ask the server or relay agent to use the broadcast IP

address for replies (a "flag" set by the client in the requests). The definition of DHCP states that implementations "should" honor this flag, but it doesn't say they "must". Some TCP/IP implementations used this flag, which meant that Relay Agents and servers had to implement it.

A number of BOOTP Relay Agent implementations (for example, in routers) handled DHCP correctly except for this feature.

6.2.3.2 Virtual LANs

With virtual LAN (VLAN) schemes that use a data packet's IP address to decide which virtual LAN a device is on, things break when DHCP is used to assign IP addresses.

DHCP servers and Relay Agents use their knowledge of what LAN the device is physically connected to select the subnet number for the device's new IP address. Whereas with switches, the subnet number sent by the device is used to decide which virtual LAN to put the device on.

6.2.3.3 Multiple Subnets on a Single Wire

Networks are sometimes configured so that one network segment has multiple subnets running to it. The DHCP server allocates addresses from a pool that is defined for each subnet or network.

In this type of environment, unless the DHCP Relay Agent is able to indicate which subnet a device requesting an address belongs to (by indicating to the server its own IP address for that subnet), or unless the server itself can be configured to cope with this environment, then you will probably not be able to implement DHCP.

6.3 IP Masquerading / Network Address Translation

One way to combat an IP address shortage is to avoid assigning addresses in the first place. Yet without an IP address no system can communicate on a TCP/IP network.

IP masquerading allows you to share one IP address among several different systems while still providing reliable, full-function TCP/IP connections to every host. This technique can be used to provide Internet service to more than one computer using a single dial-up connection from an Internet Service Provider (ISP), for example. Or you may wish to link branch office systems to your company's network without assigning IP addresses from your company's main address pool.

In fact, you still must assign an IP address to every host that needs to communicate on the TCP/IP network, but the IP addresses may be from a private network pool, such as the Class A network 10. A masquerading gateway accepts TCP/IP connection requests from these "private" hosts and translates them into requests originating from the IP address assigned to the masquerading gateway itself. Then responses are automatically routed back to the correct host on the private network. As a result, a single IP address on one network can effectively serve several different hosts without requiring additional IP address assignments on that same network.

InJoy from F/X Communications can provide IP masquerading services on your network, perhaps in combination with Virtual Private Network (VPN) capabilities. (VPN can securely link intranets at different sites using the public Internet.)

If you are making intermittent connections to the Internet perhaps NAT, Network Address Translation, is for you. For example, you have a small network at home, or you have a small office with light use of the Internet. There are a number of different products that allow you to share a modem over a regular telephone line or an ISDN line.

Sometimes NAT is referred to as IP Masquerading, although NAT is a superset of IP Masquerading (the IP address does not actually change as it passes from one LAN/WAN to another through the masquerade software). Originally NAT was suggested as a short-term solution to the problem of IP address depletion. In order to be assured of any-to-any communication in the Internet, all IP addresses have to be officially assigned by the Internet Assigned Numbers Authority (IANA). This is becoming increasingly difficult to achieve, because the number of available address ranges is now severely limited. Also many organizations have in the past used locally assigned IP addresses, not expecting to require Internet connectivity.

The idea of NAT is based on the fact that only a small part of the hosts in a private network are communicating outside of that network. So if we can devise a technique to assign official addresses to hosts that are used only when they need to communicate outside the private network, then only a small number of official addresses are required.

This is what NAT does; it takes the IP address of an outgoing packet and dynamically translates it to an official address. For incoming packets it translates the official address to an internal address. We now can use NAT for a solution for networks that have private address ranges or illegal addresses and want to communicate with hosts on the Internet.

In fact, by implementing a firewall we have already circumvented part of the problem. Clients that communicate with the Internet by using a proxy or SOCKS server do not expose their addresses to the internet, so their addresses do not have to be translated anyway. However, when we do not want, for whatever reason, to use a proxy or SOCKS server or when proxy and SOCKS are not possible, we can use NAT. For example, proxy and SOCKS servers cannot be used for UDP connections on some firewall products.

6.3.1 Translation Mechanism

For each outgoing IP packet, the source address is checked by the NAT configuration rules. If a rule matches the source address, the address is translated to an official address from the address pool. The predefined address pool contains the addresses that NAT may use for translation. For each incoming packet the destination address is checked to see if it is used by NAT. When this is true the address is translated to the original unofficial address.

Many different products implement Network Address Translation or IP Masquerading. Often true NAT is available with firewall products. One inexpensive product for OS/2 for IP Masquerading is the INJOY dialer. It also provides Dial On Demand. It can be used with modems over regular phone lines or ISDN.

For more information on IP masquerading with InJoy, please visit <http://www.fx.dk> on the Internet.

6.4 DHCP and DDNS with UNIX

UNIX has long been the standard operating system for the Internet, handling the bulk of network tasks. Most TCP/IP technologies, including the World Wide Web, grew up on UNIX. Because of this historic link to the Internet, UNIX is still one of the most popular platforms for managing a TCP/IP network.

As TCP/IP networks grow, more processing power and storage capacity may be needed to maintain DDNS and DHCP databases. In addition, the negative consequences of a network outage tend to grow as a network grows, so it's imperative that TCP/IP servers stay up and running 24 hours per day, 7 days a week. While UNIX systems may be more costly to acquire and to learn, a single network outage may be much more costly, idling many productive people. In fact, sometimes UNIX servers can replace several smaller PC servers; so even the equipments costs may be competitive.

UNIX platforms also tend to have stronger security features, which is why most large corporate firewalls run on UNIX. Most large commercial Web sites (including Yahoo and Netscape) also depend on UNIX.

UNIX may not be for every network. PC servers are increasingly getting more and more sophisticated networking technologies. (For example, Warp Server's TCP/IP 4.1 closely conforms to BSD UNIX conventions and capabilities.) However, the larger and more complex your network becomes, and the more vital your network needs, the more you should probably consider a UNIX-based TCP/IP solution such as IBM's AIX. Statistics tend to support the notion that AIX (and some other UNIX platforms) excel in reliability and performance.

The following sections describe how to set up and manage DHCP and DDNS servers running on IBM's AIX Version 4.3. We discuss security, reliability, and performance at length in following chapters. If you first need to explore these topics generally, to help assess your own network requirements, you may wish to skip ahead and then revisit these sections on IBM's AIX.

6.4.1 AIX as a DHCP Server

The DHCP configuration for AIX is held in the file `/etc/dhcpd.conf`. This file can be created and maintained either manually by using a text editor or by using the `dhcpsconf` command.

6.4.1.1 Manually Configuring the DHCP Server

The configuration file `/etc/dhcpd.conf` is a flat text file that can be maintained with any text editor, for example `vi`. For smaller locations, it is faster and easier to maintain the configuration file manually than using the GUI application.

A sample file including documentation is copied when AIX is installed.

6.4.1.2 Configuring DHCP Server through dhcpsconf

`dhcpsconf` is an X-windows GUI that lets you create and maintain configuration files and also allows you to start, stop and retrieve statistics from a running server. `dhcpsconf` also allow you to create a master definition file that can contain the configurations for all your AIX DHCP servers, and then generate individual `dhcpd.conf` files as required. Figure 173 shows the sample environment we are going to configure. It consists of two token-ring and one Ethernet network. The three networks are interconnected with two AIX boxes (buffet and banquet) that are acting as routers and DHCP servers.

There is a DNS available to all networks, and its IP address is 192.168.6.10. Devices on the 192.168.6.0 subnet include LAN workstations that must resolve NETBIOS names to IP addresses; so an NBNS is available as 192.168.6.5. (See Chapter 5, “Integrating File and Print Services” on page 177, for information on NETBIOS name resolution and NBNS)

In this network, the server buffet has DHCP configured to serve the subnets 192.168.6.0 and 192.168.9.0. Banquet will serve the subnet 192.168.7.0 and will see DHCP requests from the 192.168.6.0 subnet, but will ignore them because it contains no configuration for that subnet.

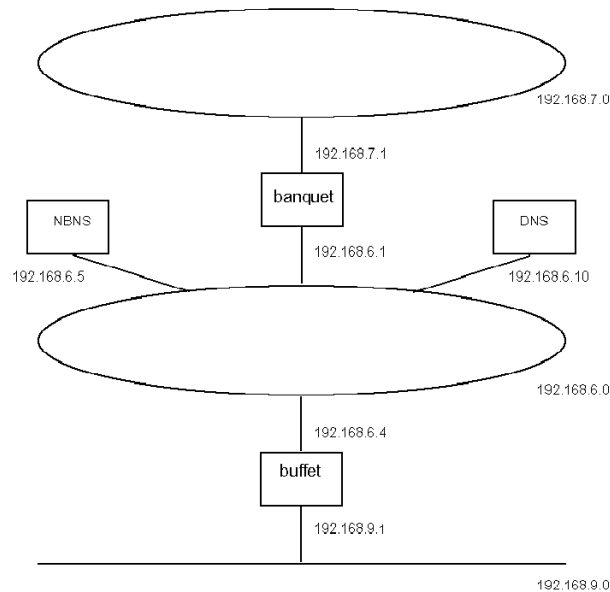


Figure 173. Sample Environment

Execute `dhcpcsf` to start the DHCP server configuration. The GUI will be opened with an empty configuration as shown in Figure 174 on page 246. We can now begin to create our master definition file.

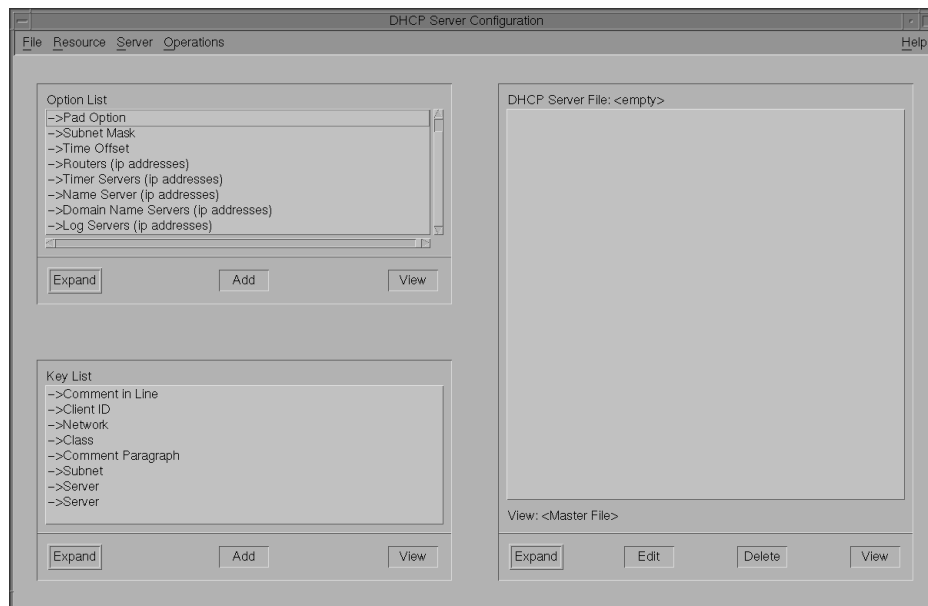


Figure 174. [AIX] DHCP Server Configuration

Add the two servers one at a time by highlighting **Server** in the Key List window and then pressing the **Add** button underneath the Key List window.

A panel like the one in Figure 175 on page 247 will be displayed. Fill in the Name and Machine name with the hostname of the server you want to configure (for clarity, we have entered the full hostname). The field names are self-explanatory.

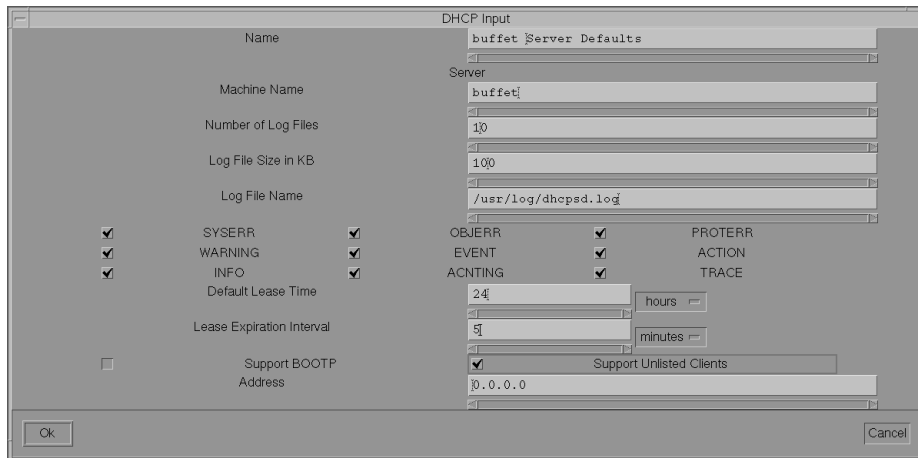


Figure 175. [AIX] Adding a DHCP Server

Change view so that we are only working with one server by selecting **[Server – View by ...]**. A selection window displaying servers in the master file will be shown. Select the server you want to work with. Notice in Figure 176 the <Master File> option. Choosing this option allows you to view the configuration for all the servers contained within the master file.

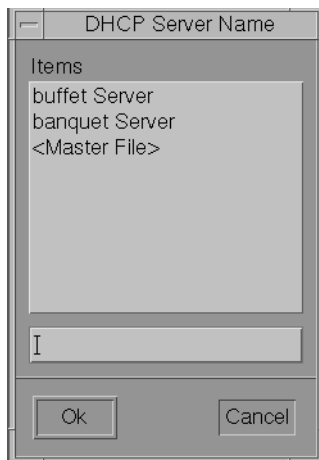


Figure 176. [AIX] Choosing a DHCP Server

We've chosen to modify the configuration for the server buffet. The first thing we're doing is adding some global options for our network. From the Option List window, we add a definitions for the domain name (austin.cooking.net)

and a DNS (192.168.6.10), as shown in Figure 177 and Figure 178 on page 248.

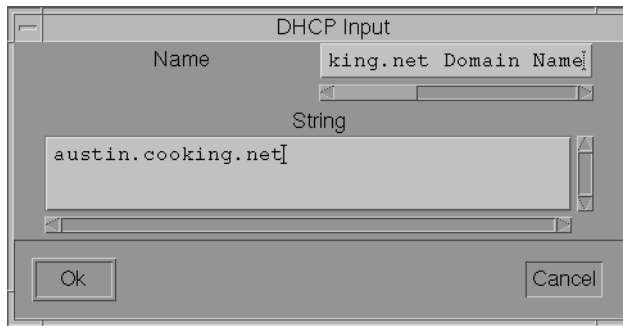


Figure 177. [AIX] Adding a Domain Name

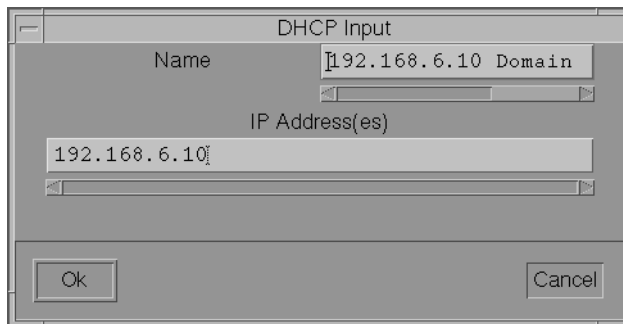


Figure 178. [AIX] Adding a DNS

Next, we'll add the 192.168.6 network. From the Key List window, add a network. The panel shown in Figure 179 on page 249 will be displayed.

As you add any item, the Name field will be filled in with a simple description and a number. The number is incremented each time you add an item and used to ensure that all the names are unique. You can change the name to whatever you wish, just as long as it is unique within your master file.

Modify the Network Address field to contain the IP address for the network. This must be a full address, for example 192.168.6.0, not 192.168.6

In the Number of bits or low end of a range field, you enter the number of bits that define the subnet mask for the network. If your mask is 255.255.255.0, then you have a 24-bit mask. Alternatively, you can enter the standard mask.

The image shows a 'DHCP Input' dialog box. It has a title bar with the text 'DHCP Input'. Inside the dialog, there are five text input fields. The first field is labeled 'Name' and contains the text '192.168.6 Network'. The second field is labeled 'Network' and is empty. The third field is labeled 'Address' and contains the text '192.168.6.0'. The fourth field is labeled 'Number of bits or low end of a range' and contains the text '24'. The fifth field is labeled 'High end of a range' and contains the text '0'. At the bottom of the dialog, there are two buttons: 'Ok' on the left and 'Cancel' on the right.

Figure 179. [AIX] Adding a Network

Notice that as you add options, they are also added to the Options List and Key List windows so that they can be added as required to the DHCP Server file window by simply highlighting them and pressing **Add**.

Now, we can add the first of our subnets. From the Key List window, add a subnet. The panel shown in Figure 180 on page 250 will be displayed.

In the Subnet Address field, enter the full IP address for the subnet, for example 192.168.6.0. In addition, add the low and high end of the range of addresses that will be served for this subnet. If you leave these fields blank, all addresses within the subnet will be served.

The DHCP Input dialog box is shown with the following fields and values:

- Name:** 192.168.6 Subnet
- Subnet:** (empty)
- Address:** 192.168.6.0
- Low end of a range:** 192.168.6.20
- High end of a range:** 192.168.6.254

Buttons: Ok, Cancel

Figure 180. [AIX] Adding a Subnet

Options can now be added to the subnet just created. Highlight the subnet in the DHCP Server file window, and choose to add options from the Option List window. In Figure 181, we're adding a default router for the 192.168.6.0 subnet.

The DHCP Input dialog box is shown with the following fields and values:

- Name:** 192.168.6.1 Router
- IP Address(es):** 192.168.6.1

Buttons: Ok, Cancel

Figure 181. [AIX] Adding a Default Router

With the addition of other subnets and options, we end up with the configuration for the buffet server that looks like Figure 182 on page 251.

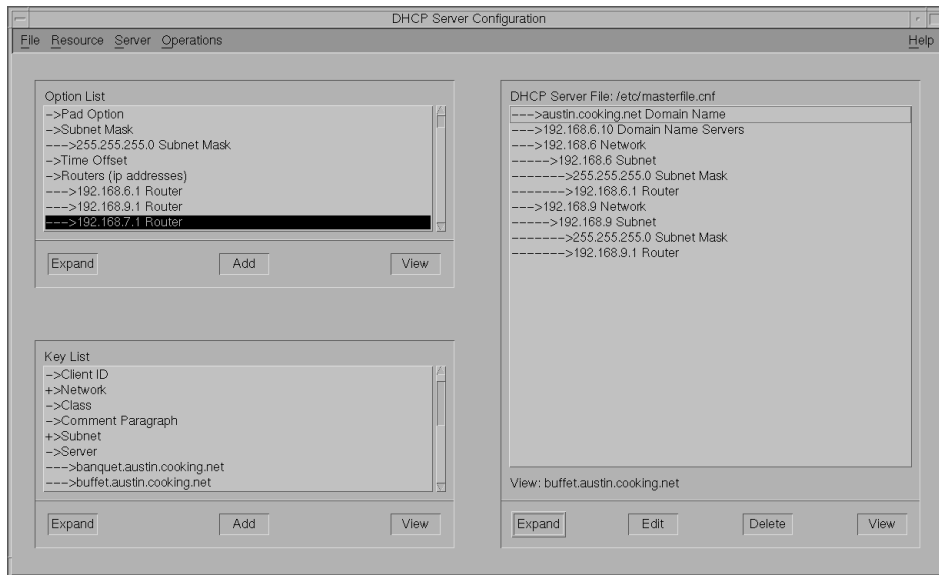


Figure 182. [AIX] Completed Server Configuration

We can now change the view to the banquet server and configure it.

Once all the servers have been configured, select [**Server — View By ... — <Master File>**]. Expand all the items in the DHCP Server file window to see your complete configuration, as shown in Figure 183 on page 252.

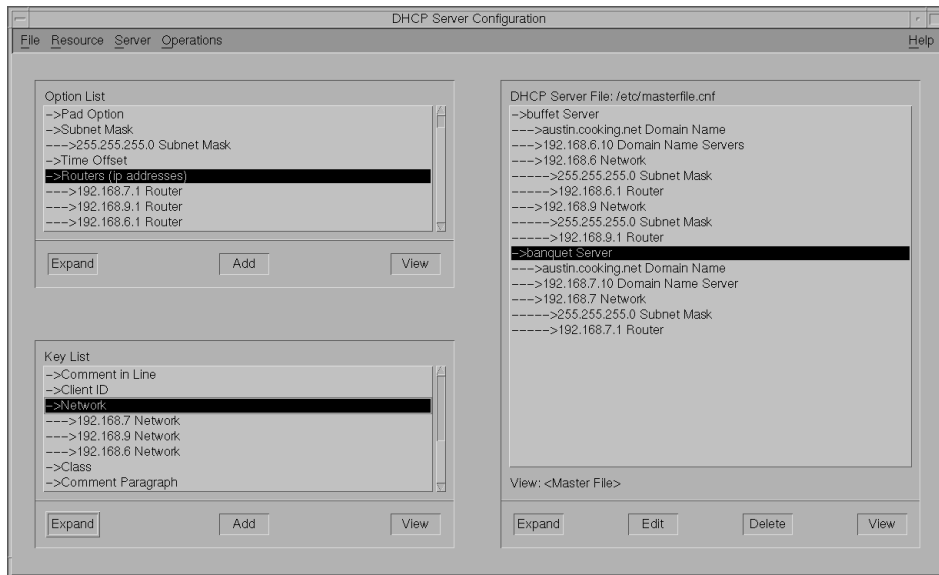


Figure 183. [AIX] Final Configuration

Select **[File — Save As]**, and change the Selection field to `/etc/masterfile.cnf` before clicking on **OK**.

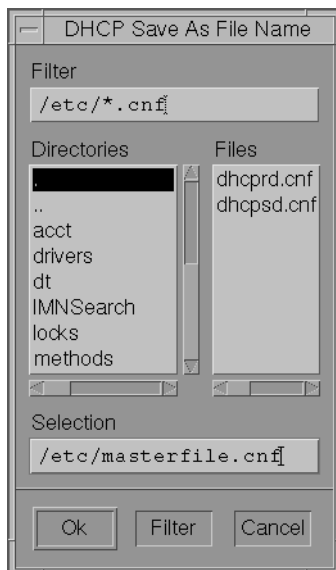


Figure 184. [AIX] Saving the Master File with [File – Save As]

To build the dhcpd.conf file for an individual server, select [**Server — Save File for**]. The server selection panel will be displayed.

Assuming we are working at the machine buffet, and we want to build its configuration file, we select **buffet.austin.cooking.net** and click on **OK**.

The DHCP Save File Name panel is displayed. Change the Selection field to /etc/dhcpd.conf, then click on **OK**.

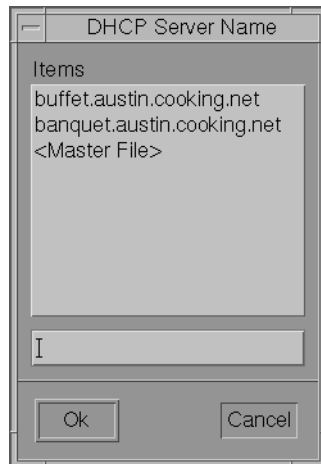


Figure 185. [AIX] Saving an Individual Server file with [Server – Save File for]

Select [**File — Quit**] to exit dhcpd.conf.

After configuration, our /etc/masterfile.conf is shown in Figure 186 on page 254.

```

### "buffet" "option 15 austin.cooking.net"
### "buffet" "option 6 192.168.6.4"
### "buffet" "network 192.168.6.0 24"
### "buffet" "{"
### "buffet" "subnet 192.168.6.0 192.168.6.20-192.168.6.254"
### "buffet" "{"
### "buffet" "option 1 255.255.255.0"
### "buffet" "option 3 192.168.6.1"
### "buffet" "}"
### "buffet" "}"
### "buffet" "network 192.168.9.0 24"
### "buffet" "{"
### "buffet" "subnet 192.168.9.0 192.168.9.20-192.168.9.254"
### "buffet" "{"
### "buffet" "option 1 255.255.255.0"
### "buffet" "option 3 192.168.9.1"
### "buffet" "}"
### "buffet" "}"

### "banquet" "option 15 austin.cooking.net"
### "banquet" "option 6 192.168.6.4"
### "banquet" "network 192.168.7.0 24"
### "banquet" "{"
### "banquet" "subnet 192.168.7.0 192.168.7.20-192.168.7.254"
### "banquet" "{"
### "banquet" "option 1 255.255.255.0"
### "banquet" "option 3 192.168.7.1"
### "banquet" "}"
### "banquet" "}"

## "austin.cooking.net Domain Name" "option 15 austin.cooking.net"
## "192.168.6.4 Domain Name Servers" "option 6 192.168.6.4"
## "192.168.6 Network" "network 192.168.6.0 24"
## "192.168.6 Subnet" "subnet 192.168.6.0 192.168.6.20-192.168.6.254"
## "255.255.255.0 Subnet Mask" "option 1 255.255.255.0"
## "192.168.6.1 Router" "option 3 192.168.6.1"
## "192.168.9 Network" "network 192.168.9.0 24"
## "192.168.9 Subnet" "subnet 192.168.9.0 192.168.9.20-192.168.9.254"
## "192.168.9.1 Router" "option 3 192.168.9.1"
## "192.168.7 Network" "network 192.168.7.0 24"
## "192.168.7 Subnet" "subnet 192.168.7.0 192.168.7.20-192.168.7.254"
## "192.168.7.1 Router" "option 3 192.168.7.1"

```

Figure 186. [AIX] The masterfile.cnf File (Part 1 of 2)

```

numLogFiles 0
logFileSize 0
logFileName
logItem SYSERR
logItem OBJERR
logItem PROTERR
logItem WARNING
logItem EVENT
logItem ACTION
logItem INFO
logItem ACNTING
logItem TRACE
leaseTimeDefault 0 minute
leaseExpireInterval 0 minute
supportBOOTP No
supportUnlistedClients No
## DEFAULTIPADDRESS 0.0.0.0
## DEFAULTHOSTNAME
### "buffet" "numLogFiles 10"
### "buffet" "logFileSize 100"
### "buffet" "logFileName /usr/tmp/dhcpd.log"
### "buffet" "logItem SYSERR"
### "buffet" "logItem OBJERR"
### "buffet" "logItem PROTERR"
### "buffet" "logItem WARNING"
### "buffet" "logItem EVENT"
### "buffet" "logItem ACTION"
### "buffet" "logItem INFO"
### "buffet" "logItem ACNTING"
### "buffet" "logItem TRACE"
### "buffet" "leaseTimeDefault 24 hour"
### "buffet" "leaseExpireInterval 5 minute"
### "buffet" "supportBOOTP No"
### "buffet" "supportUnlistedClients Yes"
### "buffet" "IPADDRESS 0.0.0.0"
### "buffet" "HOSTNAME buffet.austin.cooking.net"
### "banquet" "numLogFiles 10"
### "banquet" "logFileSize 100"
### "banquet" "logFileName "
### "banquet" "logItem SYSERR"
### "banquet" "logItem OBJERR"
### "banquet" "logItem PROTERR"
### "banquet" "logItem WARNING"
### "banquet" "logItem EVENT"
### "banquet" "logItem ACTION"
### "banquet" "logItem INFO"
### "banquet" "logItem ACNTING"
### "banquet" "logItem TRACE"
### "banquet" "leaseTimeDefault 24 hour"
### "banquet" "leaseExpireInterval 5 minute"
### "banquet" "supportBOOTP No"
### "banquet" "supportUnlistedClients Yes"
### "banquet" "IPADDRESS 0.0.0.0"
### "banquet" "HOSTNAME banquet"

```

Figure 187. [AIX] The masterfile.cnf File (Part 2 of 2)

The dhcpd.cnf file created for the buffet with the [Server — Save File for] menu options is shown in Figure 188 on page 256.

```

option 15 austin.cooking.net
option 6 192.168.6.4
network 192.168.6.0 24
{
    subnet 192.168.6.0 192.168.6.20-192.168.6.254
    {
        option 1 255.255.255.0
        option 3 192.168.6.1
    }
}
network 192.168.9.0 24
{
    subnet 192.168.9.0 192.168.9.20-192.168.9.254
    {
        option 1 255.255.255.0
        option 3 192.168.9.1
    }
}
}
numLogFiles 10
logFileSize 100
logFileName /usr/tmp/dhcpd.log
logItem SYSERR
logItem OBJERR
logItem PROTERR
logItem WARNING
logItem EVENT
logItem ACTION
logItem INFO
logItem ACNTING
logItem TRACE
leaseTimeDefault 24 hour
leaseExpireInterval 5 minute
supportBOOTP No
supportUnlistedClients Yes
## DEFAULTIPADDRESS 0.0.0.0
## DEFAULTHOSTNAME buffet.austin.cooking.net

```

Figure 188. [AIX] The *dhcpd.conf* File

6.4.1.3 Disabling BOOTP Support in *inetd*

Before the DHCP server can be started, support for BOOTP server must be removed from *inetd* because both BOOTP and DHCP use the same ports (67 and 68) to receive and send data. With BOOTP running through *inetd*, port 67 is unavailable, and *dhcpd* will not be able to start.

Issue the command:

```
grep bootps /etc/inetd.conf
```

If the line is commented (that is, the first character is a #), then BOOTP is disabled, and the DHCP server can be started.

If it isn't, you can manually edit the /etc/inetd.conf file, and comment out the line.

Alternatively, start SMIT with a `smit inetd` command. Then select **[inetd Subservers — Remove an inetd Subserver]**, and finally choose **bootps udp** from the list of Subservers that may be removed.

inetd must be restarted after modification of the configuration file. Assuming that inetd is running under SRC control, issue the command:

```
refresh -s inetd
```

If it isn't, then issue the command

```
kill -1 inetd_pid
```

Where *inetd_pid* is the pid of inetd

6.4.1.4 Starting the DHCP Server

The DHCP server subsystem can be started in several ways.

- From in `dhcpsconf`, select **[Operations — Start]**. Select the name of the DHCP server you want started.

Select **OK**.

- From a command prompt, issue the command

```
smit dhcpsd
```

Select **Start using the dhcp Subsystem**

Select **NOW**

- From a command prompt, issue the command

```
startsrc -s dhcpsd
```

6.4.1.5 Stopping the DHCP Server

The DHCP server subsystem can be stopped in several ways as well. All of these methods only stop the DHCPCD subsystem; they do not affect the rest of AIX:

- From in `dhcpsconf`, select **[Operations — Stop]**.

Select the name of the DHCP server you want stopped.

Select **OK**.

- From a command prompt, issue the command

```
smit dhcpd
```

Select **Stop Using the dhcpd Subsystem**

Select **NOW**

- From a command prompt, issue the command

```
stopsrc -d dhcpd
```

6.4.2 AIX as a DDNS Server

There are two steps required to enable DDNS on AIX.

1. At the DHCP server, add one of the following commands to the /etc/dhcpd.cnf file. (The command should be entered as a single line.)

```
updateDNS "/usr/sbin/dhchapaction '%s' '%s' '%s' '%s' PTR NONIM >>  
/tmp/updns.out 2>&1"
```

```
updateDNS "/usr/sbin/dhchapaction '%s' '%s' '%s' '%s' BOTH NONIM >>  
/tmp/updns.out 2>&1"
```

The `PTR` means that the pointer record will be updated. This is the default value. If you want to update the A record and the pointer record, then you should use `BOTH`.

2. At the DNS server (which may be the same physical box as the DHCP server), modify the /etc/named.boot file and add the `dynamic` keyword. Optionally, add the 'controlled' keyword. For example:

```
primary austin.cooking.net /etc/named.data dynamic controlled  
primary in-addr.arpa /etc/named.rev dynamic controlled
```

The `dynamic` keyword is required for the named.rev file if the `updateDNS` command is doing PTR updates. If BOTH updates are being performed, then a `dynamic` keyword is required for named.data and named .rev.

The `controlled` keyword is required for secured updates.

6.5 DHCP and DDNS with Shadow IPserver

As we have already seen in 5.4.2, "Network TeleSystems Shadow IPserver" on page 203, Shadow IPserver is a complete system for managing name and address assignments within a TCP/IP network by supporting services for DHCP, DNS, NBNS and NBDD.

See Chapter 5, “Integrating File and Print Services” on page 177, for information on integrating existing NetBIOS clients into your DHCP network, and 5.4.2.2, “Installing Shadow IPserver” on page 206 for installation instructions for Shadow IP Server.

6.5.1 Configuration Files and Save Files

Configuration files are text files manually maintained by you to hold the configuration for the DHCP, DNS and NBNS services.

The following configuration files are used by Shadow:

- NTS-SRVR.CFG** Contains system parameters such as the server's IP address, subnet mask, default router, and the IP addresses of other Shadow IP servers. This file is created at installation using information you entered during the install process.
- NTS-SRVR.DHC** Contains the configuration for the DHCP service. If you chose not to install an evaluation configuration, this file is empty.
- NTS-SRVR.DNS** Contains the configuration for the DNS service. If you chose not to install an evaluation configuration, this file contains only the address records and name records for the DNS root servers.
- NTS-SRVR.NBN** Contains the configuration for the NBNS service. If you chose not to install an evaluation configuration, this file contains and NBNS entry for the server.
- NTS-SRVR.RUR** Contains the user IDs and passwords to be used when Web security is enabled.

Only the NTS-SRVR.CFG file is required for Shadow to start.

Shadow is configured by default to create save files when it is shut down. Save files are non-editable files that contain the current configuration of the Shadow server. The configuration of your Shadow server can change during normal operation due to things it learns from clients, from other IPservers and any changes you make online by using a Web browser or an IPmanager (Such as IPmanager for Windows or IPmanager for Java).

The following save files are used by Shadow:

NTS-CFG-.SAV The save file for general system configuration and state information.

NTS-DHC-.SAV The save file for the DHCP service.

NTS-DNS-.SAV The save file for the DNS service.

NTS-NBN-.SAV The save file for the NBNS service.

If Save files are used, then when the Shadow server is started a comparison of the date and time stamps for the configuration files and their matching save files is done. The files with the later date and time stamps are used. Shadow uses the save files to restore the system and DHCP, DNS and NBNS entries to what they were when the system was shut down.

Save files can also be created at any time using either a Web browser or IPmanager. If you are using IPmanager, you can convert save files to text files and download the current server configuration as text files. See 6.5.3.6, “Backing Up Your IPserver Configuration” on page 276

6.5.2 Using Configuration Files

If you only want to use configuration files in your environment, then the `NOSAVE` parameter must be placed in the `NTS-SRVR.CFG` file. Note that disabling the creation of save files does not affect the save files that may already exist on the system; they will continue to be used as long as they are more recent than the matching configuration files.

6.5.2.1 The NTS-SRVR.CFG Configuration File

The `NTS-SRVR.CFG` configuration file is used to set global system parameters. There are over 60 keywords that may be used, and these may be split into several categories:

- Global System keywords. See Table 12.
- Global NIC keywords. See Table 13.
- Global IP keywords. See Table 14.
- Global DHCP keywords. See Table 15.
- Global DNS keywords. See Table 16.
- Global NBN keywords. See Table 17.

The global server keywords are used to define logging and save file options:

Table 12. NTS-SRVR.CFG Global System Keywords

Keyword	Description
HISTORY	Controls whether the server maintains history files.
NOLOG	Controls whether the server saves archives of the server log. The server holds four logs in memory; when the oldest log is filled, it is released.
NOSAVE	Controls whether the server creates save files.
OUTAHERE	Configures the key sequence used to stop the server. If commented out, press ESC to exit; otherwise, press leftCtrl rightShift Esc .

The NIC keywords define the network adapter installed in the server:

Table 13. NTS-SRVR.CFG Global NIC Keywords

Keyword	Description
CONNECTOR	The connector type for the 3Com adapter, either STARLAN, THICKNET, THINNET or TPI.
IO-PORT	The I/O port for the adapter.
MACADDR	The MAC address of the adapter.
NIC	The type of adapter being used, either 3C509, IBMTR or NE2000.
NO-PROMISCUOUS	Disables promiscuous mode for the 3Com adapter.
TOKENRING	Changes the network type assumed from Ethernet to token-ring.
WINDOW	The memory window used by the adapter.

The IP keywords define various TCP/IP parameters used by the server:

Table 14. NTS-SRVR.CFG Global IP Keywords

Keyword	Description
ARPTIMEOUT	The number of seconds before an arp cache entry will be discarded.
BROADCASTADDR	The type of IP broadcast to be used, either ALL-1S, Host-0S or HOST-1S.
GATEWAYADDR	The IP address of the default router.
IPADDR	The IP address of the server.
IPTIMETOLIVE	The number of seconds an IP packet can live on a network.
NETSUBNETMASK	The subnet mask of the network the server is on.
NTP-SERVER-ADDR	The IP address of an SNTP server.
TCP-IDLE-TIME	If all TCP ports are in use, this specifies the number of milliseconds a TCP port must have been idle before the server closes it for use by another client.
TCP-RETRANSMIT	How many milliseconds the server will wait for a reply before retransmitting.
TIME-OFFSET	Specifies how many half hours the server should offset the time received from the SNTP server. The offset is in an easterly direction only.

The DHCP keywords define the behavior of the DHCP service:

Table 15. NTS-SRVR.CFG Global DHCP Parameters

Keyword	Description
BOOTP	Specifies that BOOTP instead of DHCP.
DHCP	Specifies that DHCP is used. (This is the default.)
DHCPBACKUPADDR	The IP address of the DHCP Backup Peer.
DHCPBACKUPLIMIT	The number of changes made to the DHCP database before a backup packet is sent to the DHCP Backup Peer.

DHCPBACKUPTIME	The number of seconds between regular transmissions of backup packets to the DHCP Backup Peer.
PING	Makes the server ping an IP address before offering it to a client.
PING-NEGATIVE-TIMEOUT	How many seconds the server caches negative ping responses.
PING-POSITIVE-TIMEOUT	How many seconds the server caches positive ping responses.
PING-RETRIES	How many times the server should retry a ping.
REPLACE-CLIENT-ID	Allows the server to serve clients that have multiple operating systems installed, for example booting Windows 95 or OS/2. Unless you use this keyword, only the operating system that receives DHCP service first will be able to receive DHCP service again.

The DNS keywords define the behavior of the DNS service:

Table 16. NTS-SRVR.CFG Global DNS Keywords

Keyword	Description
DNSBACKUPADDR	The IP address of the backup DNS server.
DNSBACKUPLIMIT	The number of changes made to the DNS database before a backup packet is sent to the backup DNS server.
DNSBACKUPTIME	The number of seconds between regular transmissions of backup packets to the backup DNS server.
DNSCOSERVER	Specifies the IP addresses of other DNS servers (Shadow or BIND).
DNSCOSERVER-RETRIES	How many times the server will attempt to resolve a DNS query using its Coserver Peers.
DNS-COSERVER-TIMEOUT	How many milliseconds the server will wait for a reply from a DNS coserver.
DNSFORWARDER	The IP addresses of the DNS servers (Shadow or BIND) you want to use as forwarders.

DNSNAMESESERVER-RETRIES	How many times the server will attempt to resolve a DNS query using other DNS servers or DNS forwarders.
DNSNAMESESERVER-TIMEOUT	How many milliseconds the server will wait for a reply from other DNS servers or DNS forwarders.
DNSUSENBNS	Specifies whether the DNS service should query the NBNS if a DNS lookup fails. The server must be the authority for the domain in which the name being looked up resides on.
NEGCACHETTTL	How many seconds a negative reply is maintained in the DNS cache.

The NBNS keywords define the behavior of the NBNS service:

Table 17. NTS-SRVR.CFG Global NBNS Parameters

Keyword	Description
BACKUPADDR	The IP address of a NBNS Backup Peer.
BACKUPLIMIT	The number of changes made to the NBNS database before a backup packet is sent to the backup NBNS Backup Peer.
BACKUPTIME	The number of seconds between regular transmissions of backup packets to the backup NBNS Backup Peer.
COSERVER	The IP addresses of NBNS coservers.
RELEASETTL	The number of seconds before an entry marked for deletion is actually deleted.
SYNCTTL	The number of seconds a dynamic entry received from a NBNS Coserver Peer remains in the NBNS database.
TTL	The number of seconds a dynamic entry remains in the NBNS database.

6.5.2.2 The NTS-SRVR.DHC Configuration File

There are over 50 keywords that may be used in configuring the DHCP service. These keywords are used to configure Pools, Option Sets and Stations. The keywords are of two types.

First, there are the general keywords:

Table 18. NTS-SRVR.DHC General Keywords

Keyword	Description
BOOTP	When used in a pool or station definition, indicates that BOOTP should be used instead of DHCP. Using BOOTP in the NTS-SRVR.CFG file indicates that the default should be BOOTP.
DHCP	If the default is BOOTP, indicates that DHCP should be used.
ENDOPTIONSET	Ends an option-set definition.
ENDPOOL	Ends a pool definition.
ENDSTATION	Ends a station definition.
IPADDR	Specifies an IP address.
IPSOURCE	Used in pool definitions to specify the IP sources address (subnet address)
LEASETIME	Specifies the DHCP lease time.
MACADDR	Used in station definitions to specify a client's MAC address.
NO-PING	Disables (if enabled) the server DHCP ping feature.
NOT-USERCLASSID	Qualifies a DHCP client for this pool only if the client sends an ID that does not match the client ID in the pool.
OPTION	Specifies a DHCP option by number.
OPTIONSET	Begins an option set definition.
PING	Enables DHCP ping. With DHCP ping enabled, the server will attempt to ping the IP address before offering it to a DHCP client.
POOL	Begins a pool definition.
STATION	Begins a station definition.
USE	Specifies an option set to be used by a pool.

Second, there are the DHCP option keywords that allow you to specify the more commonly used DHCP options by name:

Table 19. NTS-SRVR.DHC DHCP Option Keywords

Keyword	DHCP Option	DHCP Option Name
NETSUBNETMASK	01	Subnet mask
GATEWAYADDR	03	Router
DNSADDR	06	Domain Name Server
HOSTNAME	12	Hostname
DOMAIN	15	Domain Name
IPTIMETOLIVE	23	IP Time to Live
BROADCASTADDR	28	Broadcast Address
ARPTIMEOUT	35	ARP Cache Timeout
NBNSADDR	44	NetBIOS Name Server
NBDDADDR	45	NetBIOS Datagram Distributor
NODETYPE	46	NetBIOS Node Type
LEASETIME	51	IP Address Lease Time
RENEWALTIME	58	Renewal (T1) Time
REBINDINGTIME	59	Rebinding (T2) Time
VENDORCLASSID	60	Vendor Class ID
CLIENTID	61	Client ID
NOVELL-DOMAIN	62	Novell NetWare Domain Name
NSQ-BROADCAST PREFERRED-DSS NEAREST-NWIP-SERVER AUTORETRIES AUTORETRY-SECS NWIP-1.1 PRIMARY-DSS	63	Suboptions for NOVELL-DOMAIN
USERCLASSID	77	User Class ID

The NTS-SRVR.DHC file can be created using any text editor.

Figure 189 on page 267 shows a sample NTS-SRVR.DHC for our test environment.


```
# Shadow IPserver DHCP Configuration File

POOL 'ARMONK'
  IPADDR 192.168.6.10 - 192.168.6.254, 192.168.6.11
  NETSUBNETMASK 255.255.255.0
  GATEWAYADDR 192.168.6.1
  USE 'GLOBALS'
  USE 'NETBIOS'
ENDPOOL 'ARMONK'

POOL 'REDMOND'
  IPADDR 192.168.7.10 - 192.168.7.254
  NETSUBNETMASK 255.255.255.0
  GATEWAYADDR 192.168.7.1
  USE 'GLOBALS'
ENDPOOL 'REDMOND'

STATION 'POTATO'
  IPADDR 192.168.6.11
  MACADDR 0006003912BE
  CLIENTID 'POTATO'
  USE 'GLOBALS'
ENDSTATION 'POTATO'

OPTIONSET 'GLOBALS'
  OPTION 26 576
  LEASETIME 86400
  DNSADDR 192.168.6.5
  DOMAIN 'AUSTIN.COOKING.NET'
ENDOPTIONSET 'GLOBALS'

OPTIONSET 'NETBIOS'
  NBNSADDR 192.168.6.5
  NBDDADDR 192.168.6.5
  NODETYPE P-NODE
ENDOPTIONSET 'NETBIOS'
```

Figure 189. [Shadow IPserver] NTS-SRVR.DHC File

As you can see, we have two pools called ARMONK and REDMOND. Each of these defines a subnet. Both of the pools use the options defined in the GLOBALS option set. ARMONK also uses NBNS options as defined in the NETBIOS option set.

We also have a client, POTATO (Our HP JetDirect printer discussed in 2.4.10, “Hewlett-Packard LaserJet 4000 Printer (JetDirect)” on page 70). We want POTATO to always receive the IP address 192.168.6.11 as specified with the `IPADDR` option. Seeing as this address is part of our ARMONK pool, we have explicitly excluded it from the pool by using the line:

```
IPADDR 192.168.6.10 - 192.168.6.254, 130.1.1.11
```

Anything after the comma ‘,’ is an exclusion.

6.5.2.3 The NTS-SRVR.DNS Configuration File

The NTS-SRVR.DNS file is used by Shadow to configure the DNS service. The NTS-SRVR.DNS file supplied with Shadow contains records for the well-known root servers, though you must be connected to the Internet to resolve DNS queries using these servers.

Shadow’s DNS is an implementation of RFC 1034/1035. The service uses standard DNS packets to communicate between itself and other traditional DNS servers.

If you wish to import BIND zone files into NTS-SRVR.DNS, then take note of the following:

- Only one file is used to hold the configuration.
- The format of NTS-SRVR.DNS is similar to records in BIND zone files, but all names must be fully qualified. For example, you cannot start a record with a blank or with a relative name.
- Control entries such as `$ORIGIN` and `$INCLUDE` are invalid.

When a client queries the Shadow DNS, the search is processed in the following order:

- Search the local database
- Ask Coserver Peers
- Search the local NBNS database
- Send the query to the DNS forwarders

The NTS-SRVR.DNS file can be created using any text editor.

Figure 189 on page 267 shows a sample NTS-SRVR.DHC for our test environment.

Note that the SOA record has wrapped onto the next line.

```

;# Shadow IPserver DNS Configuration File

austin.cooking.net.      IN      SOA      shadow.austin.cooking.net.
root.shadow.austin.cooking.net. (
                                1          ;Serial
                                10800       ;Refresh after 3 hours
                                3600        ;Retry every 1 hour
                                172800      ;Expire after 2 days
                                21600 ) ;Minimum TTL of 6 hours

austin.cooking.net.      IN      NS       shadow.austin.cooking.net.

shadow.austin.cooking.net.      IN      A          192.168.6.5
merlot.austin.cooking.net.      IN      A          192.168.6.10
buffet.austin.cooking.net.      IN      A          192.168.6.4
; End of File

```

Figure 190. [Shadow IPserver] NTS-SRVR.DNS File

The Shadow DNS can be integrated with an existing DNS in several ways (Depending on whether it is the authoritative server for the domain):

- The existing DNS server may be configured as a DNSCOSERVER in NTS-SRVR.CFG.
- The existing DNS server may be configured as a DNSFORWARDER in NTS-SRVR.CFG.
- The existing DNS server can have a NS record in NTS-SRVR.DNS.

6.5.3 Configuring through IPmanager

If you intend to configure your Shadow server through a Web browser or IPmanager, then you must have save files turned on. Make sure that the NOSAVE parameter in NTS-SRVR.CFG is commented out with a '#' character.

In the following sections, we will use IPmanager for Windows to administer our Shadow server.

6.5.3.1 Starting IPmanager

As you start IPmanager, a logon panel will be displayed prompting you for a user name and password. The names and passwords are stored in the NTS-SRVR.RUR file. If you have no user names set, and security is not enabled, click on **Cancel**.

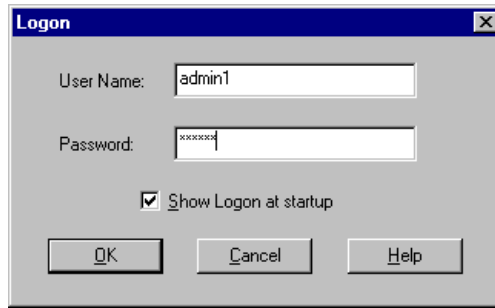


Figure 191. [Shadow IPmanager] Logon Panel

Figure 192 shows our startup screen after automatically connecting to our Shadow server. The IP address of the server can be seen in the pull-down selection box near the top-left of the screen. The logon status can be seen at the bottom of the screen.

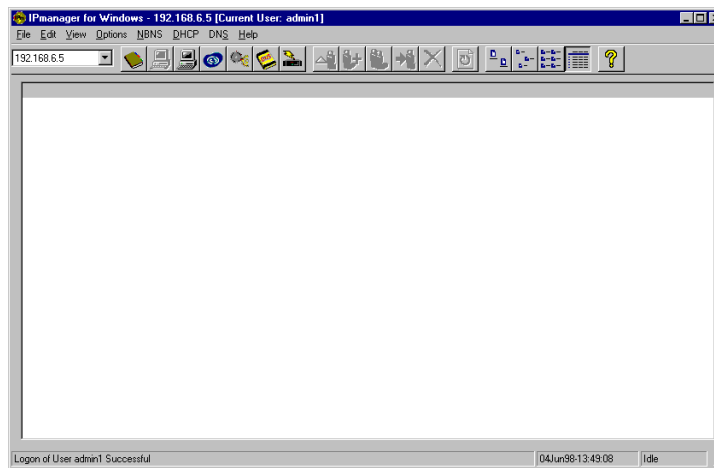


Figure 192. [Shadow IPmanager] Startup Screen

Using the pull-down menus for DHCP, DNS and NBNS, we can configure the server.

6.5.3.2 DHCP Option Sets

From the DHCP menu, select **Option Sets**. A screen showing the currently configured option sets will be displayed. Figure 193 on page 271 shows our two option sets, GLOBALS and NETBIOS.

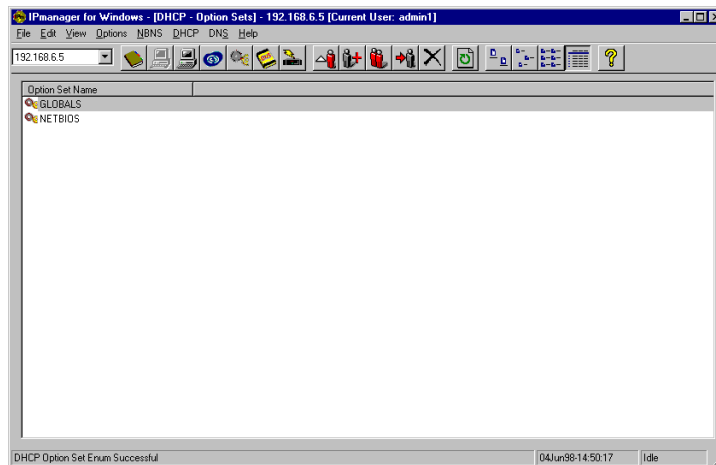


Figure 193. [Shadow IPmanager] DHCP Option Sets

You can edit an existing item by double-clicking on it and selecting **[Edit — Edit]**, by pressing **Ctrl-E**, or by using the **Edit** button on the tool bar. In Figure 194, we're editing the GLOBALS option set.

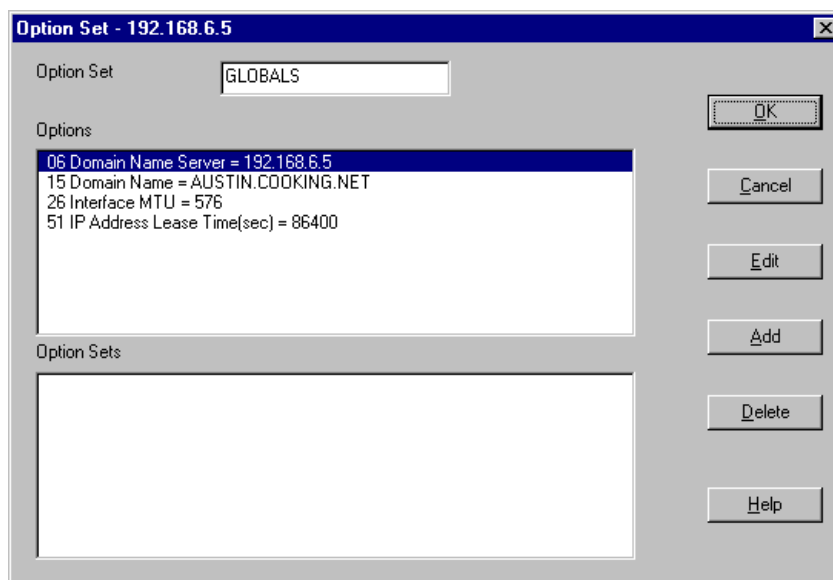


Figure 194. [Shadow IPmanager] Editing a DHCP Option Set

You can create new items by selecting [**Edit — Add**], by pressing **Ctrl-A**, or by using the tool bar. Alternatively, you can 'clone' an existing item and then modify the clone.

When you add a new option set, the edit panel (as seen in Figure 194) is used. Enter a name for the new option set in the Option Set field.

Highlight the Option field, and click on **Add** to add DHCP options to the new option set. The Add New Options panel shown in Figure 195 will be displayed. Use this to add any options to be defined in this option set.

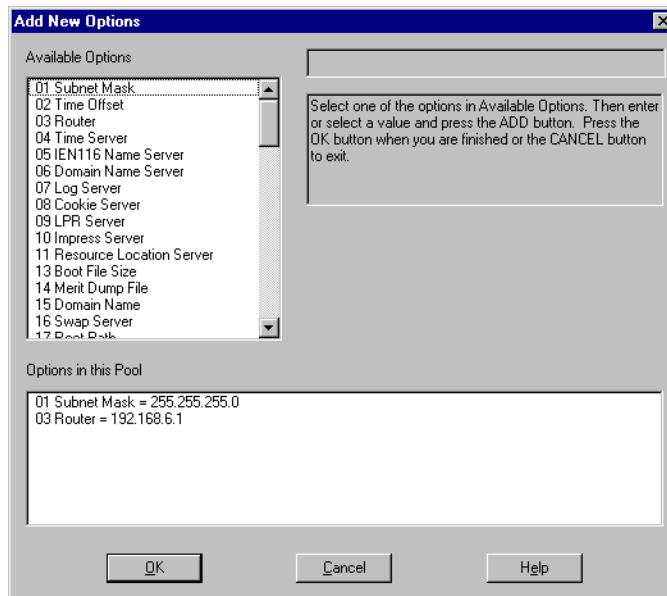


Figure 195. [Shadow IPmanager] Adding DHCP Options to a Pool

Note that option sets can include other option sets. Be aware that when nesting option sets are used, non-unique options may override one another and give unexpected results.

6.5.3.3 DHCP Pools

Select [**DHCP — Pools**] for the menu that displays the currently configured DHCP pools.

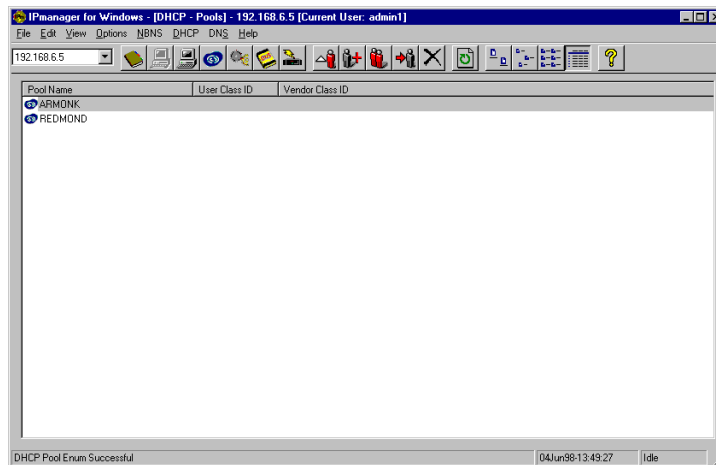


Figure 196. [Shadow IPmanager] DHCP Pools

Using the same commands that we used to maintain the option sets, you can edit, add or clone pool definitions. When maintaining pools, the panel shown in Figure 197 is used.

Note that you can add or exclude IP address ranges, individual DHCP options and option sets to a pool definition.

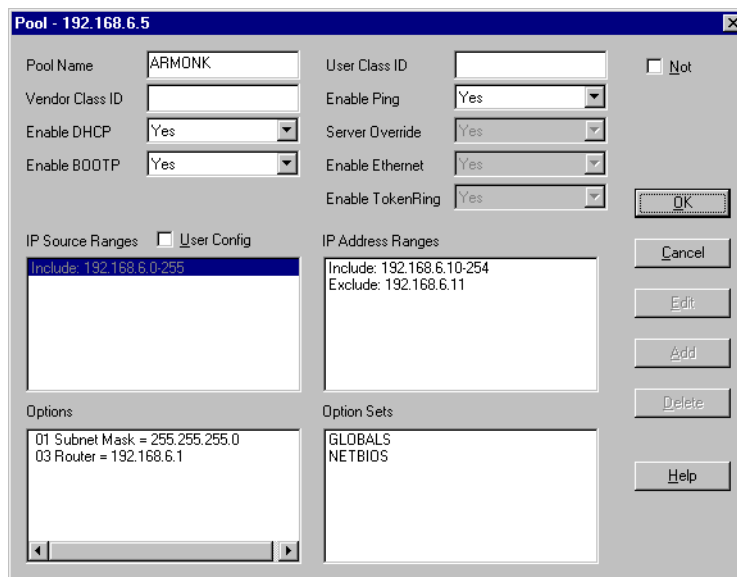
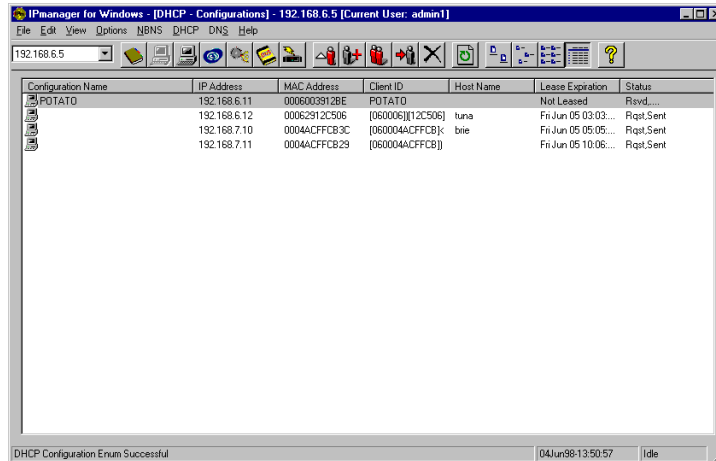


Figure 197. [Shadow IPmanager] Editing a DHCP Pool

6.5.3.4 DHCP Configurations

The **[DHCP — Configurations]** menu allows you to add individual clients to the server. You may choose to do this if you have a client that should always receive the same IP address, for example, an FTP server or a network attached printer.

Figure 198 shows the configurations on our server. It not only shows manually added clients, but also shows the clients that have received leases from the server.



The screenshot shows the IPManager for Windows interface. The title bar reads "IPManager for Windows - [DHCP - Configurations] - 192.168.6.5 [Current User: admin1]". The menu bar includes File, Edit, View, Options, NBNs, DHCP, DNS, and Help. The toolbar contains various icons for configuration management. The main window displays a table with the following data:

Configuration Name	IP Address	MAC Address	Client ID	Host Name	Lease Expiration	Status
POTATO	192.168.6.11	00060039128E	POTATO		Not Leased	Rev'd...
	192.168.6.12	00062912C506	[600006][12C506]	tuna	Fri Jun 05 03:03:...	Rqst.Sent
	192.168.7.10	0004ACFFCB3C	[600004ACFFCB]<	brle	Fri Jun 05 05:05:...	Rqst.Sent
	192.168.7.11	0004ACFFCB29	[600004ACFFCB]		Fri Jun 05 10:06:...	Rqst.Sent

The status bar at the bottom indicates "DHCP Configuration Enum Successful" and the time "04Jun98:13:50:57".

Figure 198. [Shadow IPmanager] DHCP Configurations

The Configuration panel shown in Figure 199 on page 275 is used to maintain individual DHCP client configurations.

Configuration - 192.168.6.5

Configuration: IP Address: ☒ Configured

Lease Expires: MAC Address: ☒ Configured

Enable DHCP: Client ID: ☒ Configured

Enable BOOTP: Enable Ping:

Configured Options:

Server Override:

Enable Ethernet:

Enable TokenRing:

Option Sets:

Assigned Options:

Incoming Options:

Additional Information:

Lease None

Reserved

Backup Complete

State	Giaddr	XID	Last Packet	Status
<None>	<None>	00000000	<None>	<Rsvd>,<....>

Figure 199. [Shadow IPmanager] Editing a DHCP Station

6.5.3.5 DNS Entries

The **[DNS — DNS Entries]** menu shows the current DNS database. Note that in Figure 200 we have expanded the tree so that the entries in the austin.cooking.net domain are displayed.

The DNS database not only includes any manually configured names, but also includes dynamic updates to the DNS database.

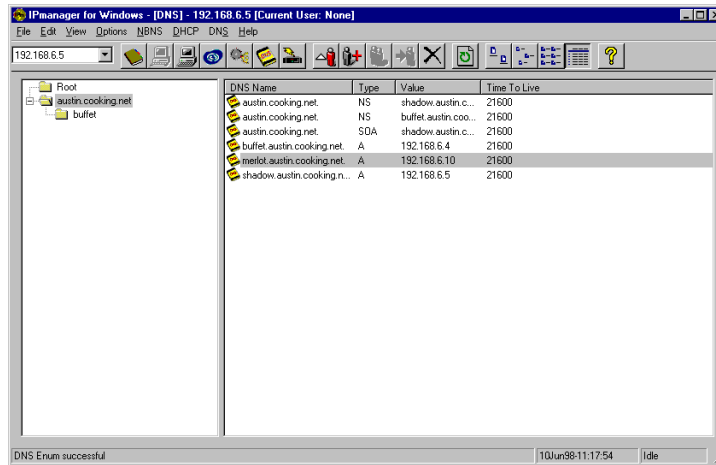


Figure 200. [Shadow IPmanager] DNS Entries

Figure 201 shows the panel used to add an entry to the DNS database. As you can see, you can add standard record types such as A and CNAME.

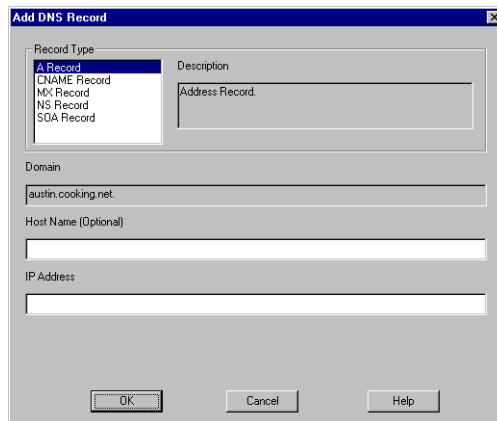


Figure 201. [Shadow IPmanager] Adding a DNS Entry

6.5.3.6 Backing Up Your IPserver Configuration

Of course, once you have made all the modifications you require to the Shadow configuration, you want to save it. Rather than shutting down the server to save the save files, you can select the **[Options — Server Save]** menu to write the save files.

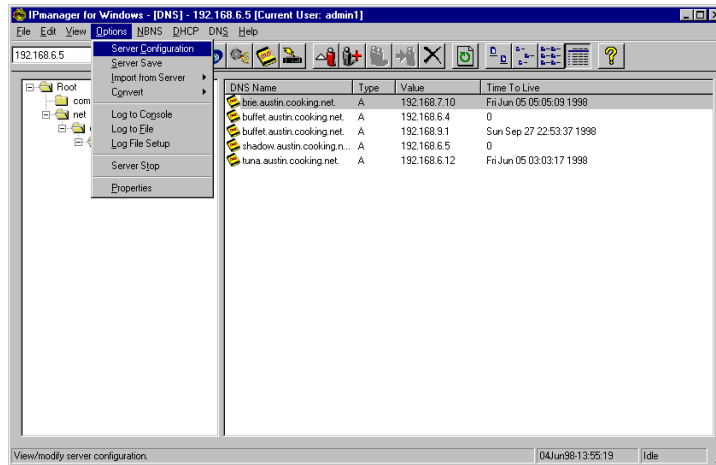


Figure 202. [Shadow IPManager] The Options Menu

If you are using a Web browser to administer your Shadow server, you can also force a write of the save files.

Using the [**Options — Import from Server**] menu as shown in Figure 203, you can download the current configurations as a text files to your local machine.

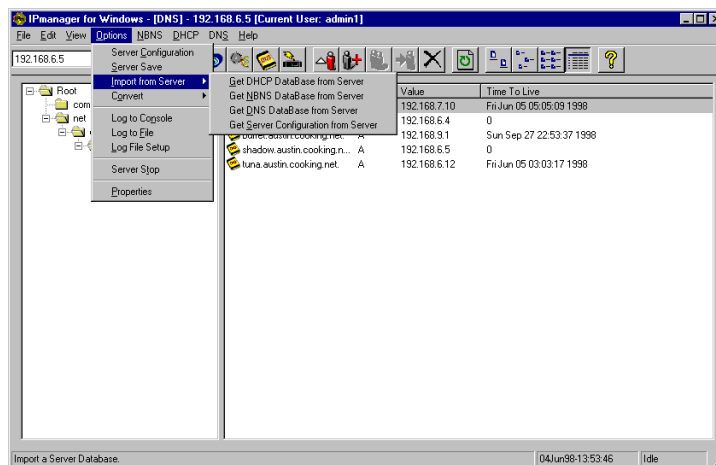


Figure 203. [Shadow IPManager] Importing Server Databases

Using the [**Options — Convert**] menu as shown in Figure 204 on page 278, you can convert the non-editable save files to text files.

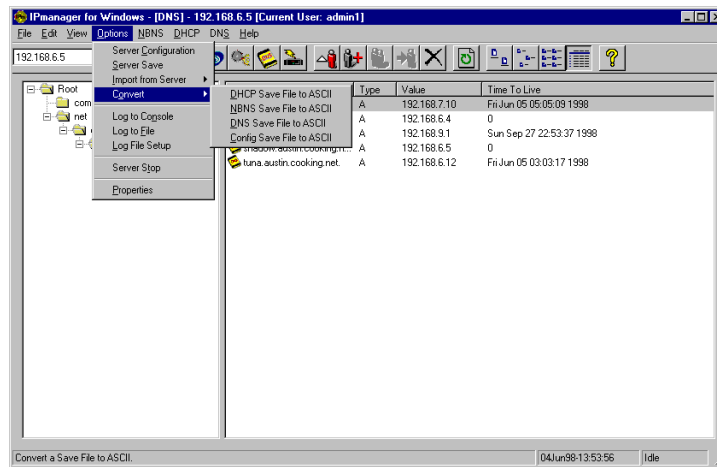


Figure 204. [Shadow IPmanager] Converting Save Files to Text

Chapter 7. Mobile Users

If you're planning on supporting remote users or offices, this chapter will help explain PPP with dial-up or ISDN connections. We'll also demonstrate how to automatically assign printers and display welcome messages as users roam from office to office.

7.1 PPP Dial-Up

PPP servers provide dial-up access to a TCP/IP network over modem, ISDN, or similar connections for remote users working away from the office. Sometimes these servers also link remote branch offices to the central office. Unfortunately, most PPP servers must be managed separately, and, while they can provide IP addresses to clients dynamically, they usually supply those addresses out of separate, static tables maintained at each PPP server. Consequently, they do not integrate well with DHCP, and they can impose an extra burden on network managers.

However, IBM's PPP Server for OS/2 Warp Server, available through the IBM Software Choice program, can access a DHCP server to obtain addresses for remote users. (See Figure 205 and Figure 206 on page 281.) DDNS services may also be used. By shifting the burden for address assignment back to the DHCP server(s), the IBM PPP Server can prove a lot easier to administer.

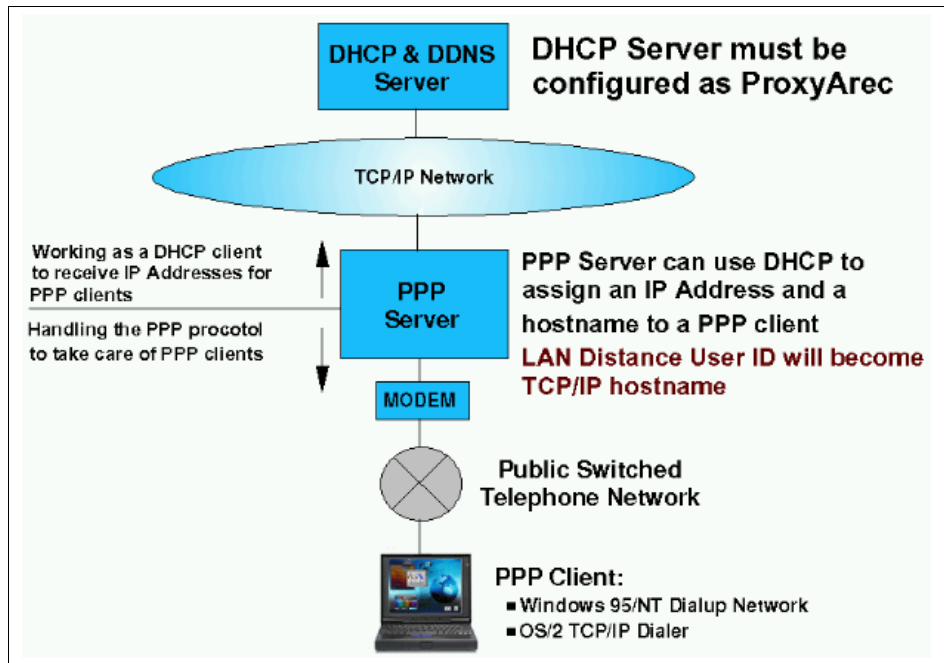


Figure 205. [Warp Server] PPP Server and DHCP

7.1.1 Prerequisites

To install an IBM PPP Server on your network, you will need the following:

1. A server with OS/2 Warp Server installed. (You do not necessarily need the "Advanced" version of OS/2 Warp Server for this task since it is generally not disk intensive.) The server should have at least one dial-up connection available (such as a modem) for testing purposes and should also be connected to your TCP/IP network.

2. The IBM PPP Server software from the IBM Software Choice Web site at:

<http://www.software.ibm.com/swchoice>

You will need a Software Choice or Passport Advantage subscription, available from your IBM software dealer, to access this software.

3. If you plan to use TCP/IP 4.1 with the PPP Server, you should also obtain at least two fixes: APARs IC21069 and IC21116. These fixes are available from IBM Software Support on the Internet at:

<http://ps.software.ibm.com>

4. A DHCP server on your network, such as OS/2 Warp Server's TCP/IP 4.1 DHCP server. (In fact, the PPP Server can reside on the same system as the DHCP server.)
5. A DDNS server if you wish to take advantage of dynamic name services. (Optional.)
6. A remote client with a PPP dialer for testing purposes.

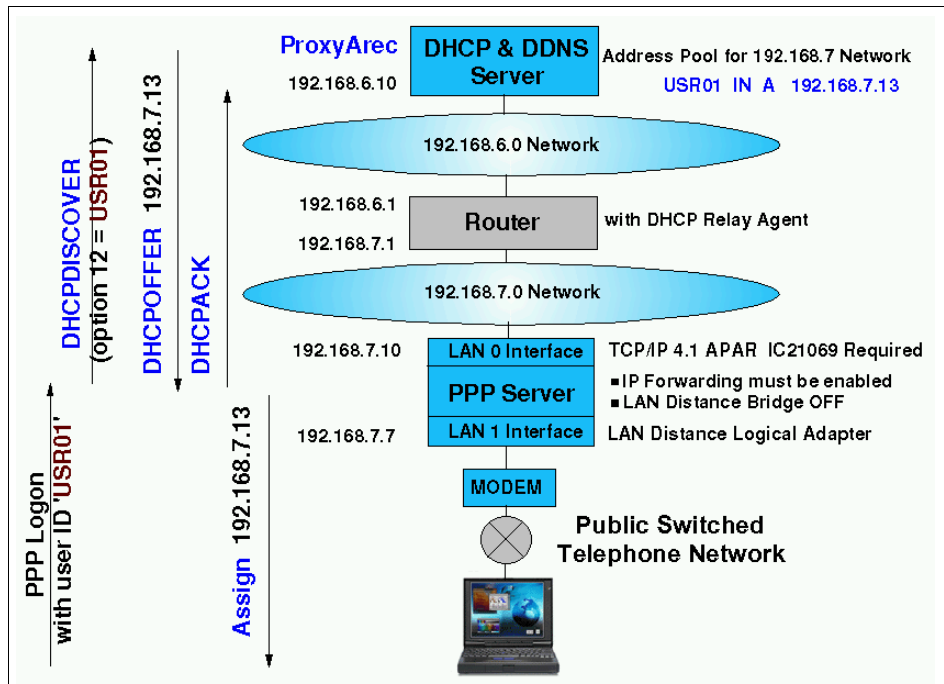


Figure 206. [Warp Server] PPP Server and DHCP: Example

The IBM PPP Server (formerly called LAN Distance) works with many, but not all, network adapters supported by OS/2 Warp Server. Choose the network adapter carefully for your PPP server. A list of supported adapters can be found on the Internet at the OS/2 Device Driver Pak On-Line at:

<http://service.software.ibm.com/os2ddpak/index.htm>

With the correct communications adapter, the IBM PPP Server can support up to 128 dial-up ports per server using products such as NETAnywhere. For more information on NETAnywhere, please visit:

<http://vrcomm.com/html/vcpos2.html>

7.1.2 Step-by-Step Procedure

Here are the major steps required to set up the IBM PPP Server with DHCP:

1. *Install the IBM PPP Server software.* Simply follow the normal procedure described in the documentation.

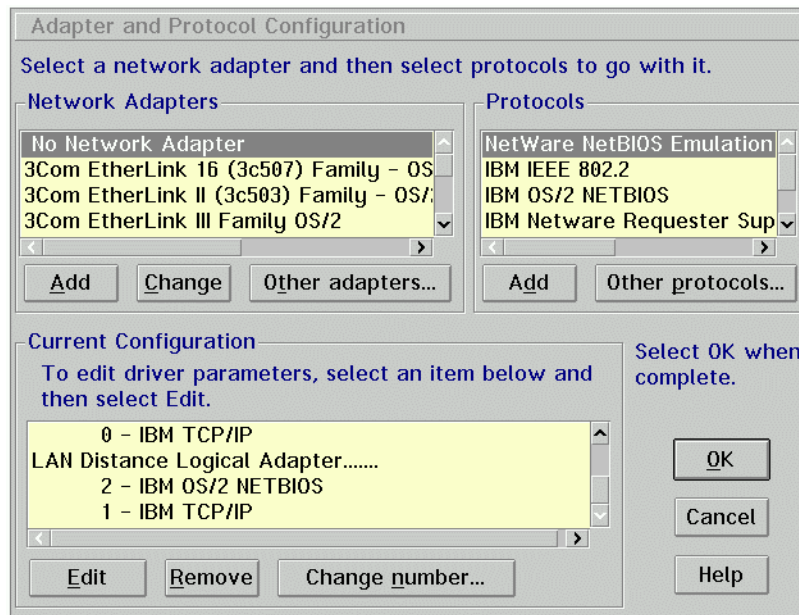


Figure 207. [Warp Server] TCP/IP Added to PPP Logical Adapter

2. *Verify that the TCP/IP protocol is bound to both adapters.* Using MPTS, make sure that the TCP/IP protocol is listed under both network adapters. "0 - IBM TCP/IP" should be listed under the regular LAN adapter and "1 - IBM TCP/IP" should be listed under the LAN Distance Logical Adapter. See Figure 207.
3. *Configure IP addresses for both adapters.* Note that these addresses need not be on the same subnet. Although DHCP can be used to assign these addresses, you may wish to stick to static addresses until your other testing has been completed.
4. *Turn on IP forwarding.* TCP/IP traffic must be allowed to flow from the LAN Distance Logical Adapter to your regular LAN adapter and out onto the network. Open the TCP/IP Configuration notebook and check the **IP**

Forwarding checkbox. See Figure 208.

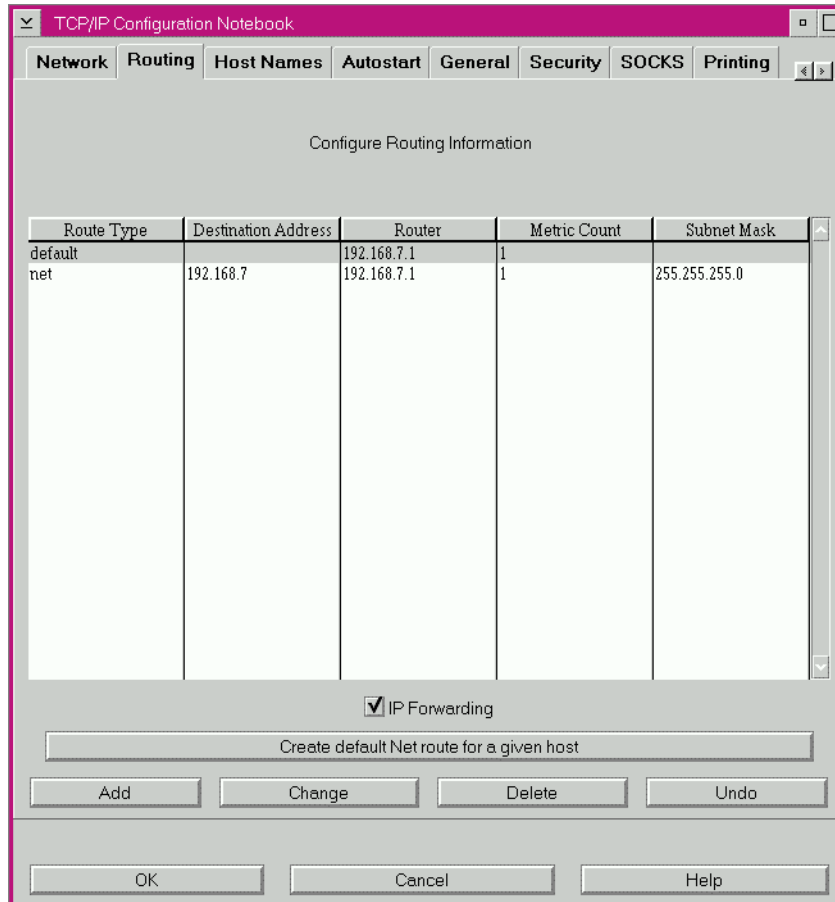


Figure 208. [Warp Server] IP Forwarding Enabled

5. *Edit WCLLOCAL.INI to turn on DHCP.* In the \WAL directory, a key configuration file for the PPP Server, WCLLOCAL.INI, can be found. Using any text editor, such as the OS/2 System Editor, find the [PPP] section and change the OBTAINIPADDR line so that it reads:

OBTAINIPADDR=DHCP

(You may also wish to activate DDNS.)

6. *Define an address pool for PPP clients on the DHCP server.* The subnet containing the dial-up PPP clients should have enough free addresses available.

7. *Define a PPP user.* Start the IBM Remote Access program. Using this graphical program, define at least one PPP user for testing.
8. *Add the PPP Server to the Startup folder.* Drag a shadow of the IBM Remote Access program icon to the Startup folder to make sure that the PPP Server always starts up every time the server reboots.
9. *Test the dial-up connection.* Your dial-up PPP client should be able to connect to the server, get an address assignment, and communicate with other systems using the user ID and password you defined earlier. External modems on both ends can prove helpful since they provide status indicators which can be used to monitor the progress of connections.

The online PPP Server reference guide includes more information on troubleshooting your dial-up connections. In particular, you should pay close attention to the routing tables and subnets you have implemented on your network, since most common errors occur when dial-up clients do not have proper routing to other systems on the network.

7.2 Roaming Users

While we've seen how people can work at home through a dial-up PPP connection, and get access to the office TCP/IP network, it isn't the only place where dynamic network access may be needed. Many companies have several different locations and workers who travel may need to connect from a remote office. It's important that they feel at home, getting not only IP address information but also access to local network printers and any other resources particular to that specific office. That way, these travelers can get to work as quickly and easily as possible.

Sales people, service personnel, executives, and many other professionals must work in several different locations from time to time, and many of them rely heavily on their notebook computers. "Hotelling" is also becoming commonplace, with telephone systems and computing technologies that allow employees to sit at any desk in the office. A dynamic IP network can help respond to these changing business needs. This section focuses on extending DHCP to help assign local printers, display a local "message of the day," and otherwise provide "no brainer" access to the network wherever a roaming user may be.

For this section, we tested OS/2 Warp 4 on an IBM ThinkPad 760ED notebook computer with the IBM PCMCIA Token-Ring Adapter. However, you may adapt these scenarios to your own particular software, equipment, and

network. In fact, since almost every company has roaming users of one sort or another, we believe most readers ought to study this section carefully.

7.2.1 Prerequisites

To provide services for roaming users, you will first need access to the following resources:

1. A DHCP server (preferably with DDNS), such as OS/2 Warp Server. We recommend you set a relatively short lease time (such as three minutes; three minutes is the shortest lease time allowed by the DHCP server) for testing purposes.
2. A sample client system, with basic DHCP capabilities (and, if necessary, file/print services with TCPBEUI) working correctly.
3. Access to at least one other location, whether it's across the hall, another floor, or in a different building. This location may be on a different subnet.
4. A Web server, such as Lotus Domino, for "message of the day" service.
5. A network printer, on a OS/2 Warp Server or NT domain, or as a TCP/IP (LPD) printer.

7.2.2 Automatic Connection to File/Print Domain

If you primarily use traditional file and print services, reserving TCP/IP for Web browsers and other applications, then you can use extended DHCP options to pass information on the local file/print domain to a roaming client.

7.2.2.1 Assumptions

For now, we assume:

1. A roaming user visits at least two different offices, floors, or other separate file/print domains with a notebook computer running OS/2 Warp 4.
2. The roaming user has an account on each domain (even a GUEST account) with default printer assignments.
3. The domain controller and the client can use TCPBEUI to communicate.
4. For simplicity, the domain controller is doubling as the print server, and the client does not need to use TCPBEUI to communicate with any other systems. In other words, only one TCPBEUI broadcast address is needed by the client.
5. A NetBIOS name server, such as Shadow IPserver, is not available.

Both the domain name and the IP address of the domain controller may vary from location to location.

7.2.2.2 DHCP Server Setup

DHCP provides some standard options (44 to 47) for passing NetBIOS-related information to clients. However, none of these options seem appropriate given our assumptions. For example, there's no explicit option to provide the domain name to our client. While it's possible to use a published DHCP option for something other than its specified use, we certainly don't recommend the practice.

Instead there are at least two ways to extend DHCP to provide this custom information to each client. One way is to use Option 43 ("Vendor Specific Information"). Perhaps the best way, though, is to use any of the options already set aside for custom use, options 128 through 254.

Although not universally accepted, many platforms have settled on specific purposes for some of the DHCP options in this range. OS/2 Warp, for example, may use options 200 to 208 to receive various extra TCP/IP parameters, such as the default SOCKS server address (Option 205). Consequently, these should be avoided as well when choosing your own custom options.

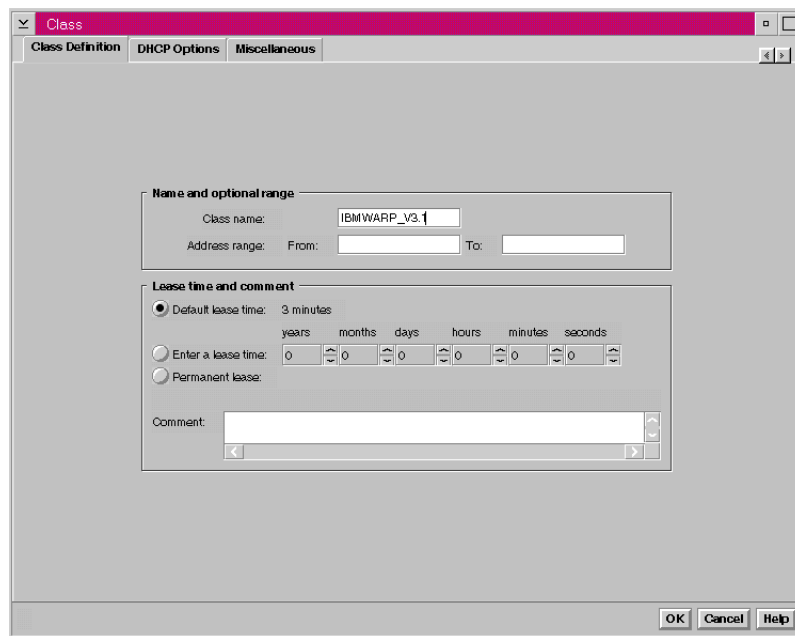


Figure 209. [Warp Server] Adding Class Definition to DHCP Server

We need to add two extra, custom options to the DHCP server. The following two options can be defined:

- Option 150 Domain Name
- Option 151 TCPBEUI Broadcast Address

Figure 210. [Warp Server] Adding Option 150 to Class IBMWARP_V3.1

For our first test office, the domain name is ARMONK, and the TCPBEUI broadcast address is merlot.armonk.cooking.net. Since these are not reserved DHCP options, they shouldn't bother any other DHCP clients; so you can safely add them to any subnet you wish. Try adding them to your DHCP server for a specific subnet, client classification, or other portion of your network where they would apply.

In Figure 209, we define a new class, IBMWARP_V3.1, for the entire subnet, armonk.cooking.net. The class name, IBMWARP_V3.1, is found in the DHCP.D.CFG file on an OS/2 Warp client, and it's normally the default value for both options 60 and 77. (You can edit DHCP.D.CFG on the client if you wish to change this value.) This class name is communicated to the DHCP server at the time the client requests its address assignments. In this example, listing options 150 and 151 under a specific class name prevents other clients, even within the same subnet, from receiving these extra option values.

Figure 210 and Figure 211 show how one of these extra options, the LAN Domain Name, is defined and its value, ARMONK, set.

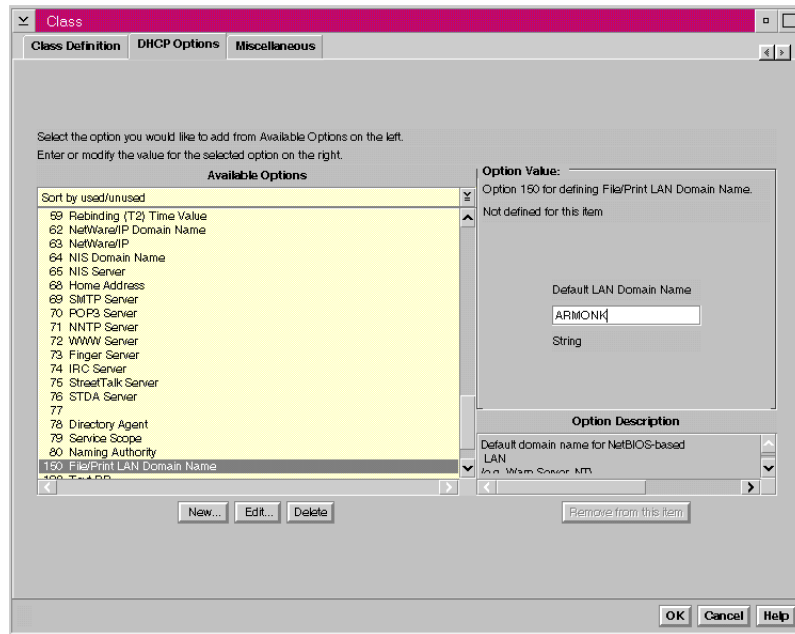


Figure 211. [Warp Server] Setting ARMONK as DHCP Option 150 Value

Finally, in Figure 212, all configuration is completed. With our new class highlighted, IBMWARP_V3.1, the options which will be received by members of this class are listed, with their values, on the right. In particular, option 150 will provide the string ARMONK, and option 151 will provide the fully qualified domain name for the domain controller, the string merlot.armonk.cooking.net. In fact, merlot.armonk.cooking.net is registered with a Dynamic DNS server, and its numeric IP address assignment could change; so it's particularly important that a name server be provided to the client as one of the options to help properly resolve this TCP/IP address.

Although options 44, 45, and 46 are defined for this entire subnet, they will not be used by this client. We assume that a NetBIOS name server is not available.

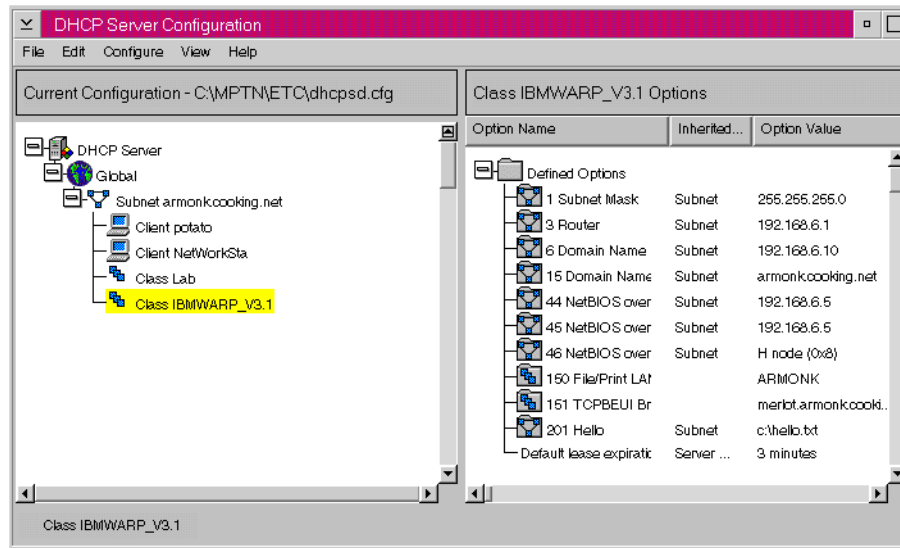


Figure 212. [Warp Server] Option Values for Class IBMWARP_V3.1

7.2.2.3 Client Setup

Our client (OS/2 Warp 4 in this example) has been previously set up to get its IP address assignment using DHCP and to use TCPBEUI as the protocol for its File/Print Client software. Now it's up to OS/2 Warp 4 to do something with these two new DHCP values.

In OS/2 Warp 4, the file DHCPDC.CFG controls the DHCP client configuration. This file is located in the directory pointed to by the ETC environment variable, as set in the CONFIG.SYS file. Normally, but not always, this directory is X:\MPTN\ETC (where "X:" is the OS/2 Warp boot drive). You can edit this file with any text editor, such as the OS/2 Warp System Editor.

Part of the DHCPDC.CFG file describes the DHCP options the client should handle (and how they should be handled). That section appears below:

```
#option 9   exec "dhcpihm.cmd 9 %s"      # LPR Server
#option 71  exec "dhcpihm.cmd 71 %s"     # Default NewsReader/2
#option 200 exec "dhcpihm.cmd 200 %s"    # Default LPR Printer
#option 201 exec "dhcpihm.cmd 201 %s"    # Gopher Server
#option 202 exec "dhcpihm.cmd 202 %s"    # Default WWW Home Page
[...]
```

The "#" symbol indicates that the option is commented out and deactivated. We need to add options 150 and 151 and process them appropriately. So, in this section, add the following lines (without the "#" symbol in front):

```
option 150 exec "myoffice.cmd 150 %s" # Domain Name
option 151 exec "myoffice.cmd 151 %s" # TCPBEUI Broadcast Address
```

By adding these lines to the DHCPDC.CFG file, we're instructing OS/2 Warp 4 to obtain these extra option values (150 and 151) from the server and to pass them to a REXX script called MYOFFICE.CMD, which you can create. MYOFFICE.CMD will be run twice, once for each option. Unless the directory is specified (for example, C:\BATCH\MYOFFICE.CMD), the PATH (as set in CONFIG.SYS) must contain the directory where MYOFFICE.CMD is located.

DHCPIBM.CMD is provided as a sample REXX script to show how DHCP options can be processed. You can find DHCPIBM.CMD in the \MPTN\BIN directory.

Here's how MYOFFICE.CMD might look, using DHCPIBM.CMD as a basis:

```
/* MYOFFICE.CMD
 * by Timothy Sipples
 * MYOFFICE <option tag> <option data>
 * Return values: 0=successful, 1=input error, -1=system error
 */

/* add REXXUtil functions */
call RxFuncAdd 'SysFileDelete', 'RexxUtil', 'SysFileDelete'

return_code = 0
bcst_file = "C:\IBMCOM\RFCBCST.LST" /* Drive C; change if needed */

/* Get option number and data from the command line */
parse arg option_tag option_data

if ((option_tag = '') | (option_data = '')) then do
  return_code = 1 /* wrong number of arguments */
  exit return_code
end

/* Strip leading and trailing spaces */
option_tag = strip(option_tag)
option_data = strip(option_data)
```

Figure 213. [OS/2 Warp] MYOFFICE.CMD (Part 1 of 3)


```

/* Instantiate the option */
select
  when (option_tag = 150) then do      /* Domain Name */
    data = hex2ascii_string(option_data)
    if (data \= '') then return_code = change_domain(data)
    else return_code = 1      /* improper Domain Name */
    end

    when (option_tag = 151) then do      /* TCPBEUI B'cast Address */
      data = hex2ascii_string(option_data)
      if (data \= '') then do
        dummy=SysFileDelete(bcst_file)
        return_code = lineout(bcst_file,data)
        dummy=stream(bcst_file,"C","CLOSE")
        "@rfcaddr"
      end
      else return_code = 1      /* improper B'cast Address */
      end

    otherwise
      return_code = 1
    end

exit return_code

/* hex2ascii_string (option_data)
 * Take a hex string in the form "nn nn nn ...", verify it's a valid
 * hex string, and convert it to an ASCII character string.
 * Returns the string if the data is valid, a null string if not.
 */
hex2ascii_string: procedure
  parse arg 'hex' '' data ''
  if (\ DATATYPE(data, X)) then return ""
  data = strip(data, 'B', '09'x)
  return X2C(data)

/* change_domain (domain_name)
 * Take the domain_name and record it in IBMLAN.INI
 */
change_domain: procedure
  parse arg domain_name
  lanini_file = "C:\IBMLAN\IBMLAN.INI"      /* Change if needed */

```

Figure 214. [OS/2 Warp] MYOFFICE.COMD (Part 2 of 3)

```

temp_file = "C:\IBMLAN\IBMLAN.TMP"      /* Change if needed */
dummy = SysFileDelete(temp_file)
dummy = stream(lanini_file,"C","OPEN READ")
dummy = stream(temp_file,"C","OPEN WRITE")
do while lines(lanini_file)
    temp_line = linein(lanini_file)
    if left(translate(strip(temp_line)),6) = "DOMAIN" then
        temp_line = "  Domain = "||domain_name
        dummy = lineout(temp_file,temp_line)
    end
dummy=stream(lanini_file,"C","CLOSE")
dummy=stream(temp_file,"C","CLOSE")
dummy=SysFileDelete(lanini_file)
"@copy "temp_file" "lanini_file
return 0

```

Figure 215. [OS/2 Warp] MYOFFICE.CMD (Part 3 of 3)

When DHCP option 150 is received by MYOFFICE.CMD, the domain name is extracted and recorded in the IBMLAN.INI file. When DHCP option 151 is received, MYOFFICE.CMD extracts the broadcast address and creates a new RFCBCST.LST file containing that address. Then RFCADDR runs and rereads the broadcast and names files (RFCBCST.LST and RFCNAMES.LST) for TCPBEUI.

When the desktop loads, STARTUP.CMD works in partnership with MYOFFICE.CMD. Here's how STARTUP.CMD may look (Figure 216 on page 293):

```

/* STARTUP.CMD
 * by Timothy F. Sipples
 * Waits for MYOFFICE.CMD to rewrite IBMLAN.INI then proceeds
 * to start requester and prompt for logon.
 */

lanini_file = "C:\IBMLAN\IBMLAN.INI" /* Change if needed */

call RxFuncAdd "SysFileTree","RexxUtil","SysFileTree"
call RxFuncAdd "SysSleep","RexxUtil","SysSleep"

"@cls"
say "Waiting to receive your local domain name."
say "One moment, please..."

do forever
  dummy = SysFileTree(lanini_file,"info","T")
  curr_date = date("O")
  curr_time = time("M")
  parse value info.1 with f_yy "/" f_mo "/" f_dd "/" f_hh "/" f_mm " " .

  file_date = f_yy||"/"||f_mo||"/"||f_dd
  file_time = f_hh*60 + f_mm

  if curr_date = file_date then
    if (curr_time = file_time) | (curr_time = file_time+1) then leave

  if (curr_time = 0) & (file_time = 60*23+59) then do
    /* Insert midnight rollover handling here. */
  end

  call SysSleep 10
end

"@START /N /PM LOGON /V:DOMAIN"
"EXIT"

```

Figure 216. [OS/2 Warp] STARTUP.CMD

STARTUP.CMD waits for MYOFFICE.CMD to rewrite IBMLAN.INI with the correct default domain name. Then STARTUP.CMD starts LOGON, which first starts LAN Requester. The user can then log onto the local domain to obtain default printer assignments and other LAN resources.

STARTUP.CMD knows that MYOFFICE.CMD has finished with IBMLAN.INI by comparing the file date and time with the current date and time. If these two times differ by no more than one minute, MYOFFICE.CMD has likely done its work, and the LAN Requester starts. If not, STARTUP.CMD keeps checking every 10 seconds. Since MYOFFICE.CMD could take up to a minute or more to get the extra DHCP options, STARTUP.CMD warns the user with a message, *One moment, please...*

7.2.2.4 Technical Caveats

You should view MYOFFICE.CMD and STARTUP.CMD as examples to illustrate the capabilities of DHCP in helping to support mobile users. Many potential problems exist in both batch files. We would like to draw your attention to a few of these issues:

- MYOFFICE.CMD runs with every lease renewal, once for each of the two new DHCP options. If the lease period is short, then a lot of background processing can take place, impacting performance. Also, tasks which require user input (such as LOGON) should not be started from MYOFFICE.CMD or any equivalent. If the lease time is two minutes, for example, then the user would be prompted every two minutes to log onto the network. Use the DHCP option handler to record option values in INI files, configuration files, and other locations. Use other batch files (such as STARTUP.CMD or a program started from the Startup folder) to act on these option values.
- We believe that it's appropriate to delete RFCBCST.LST and rewrite the file with each renewal. A more sophisticated version of MYOFFICE.CMD should perhaps modify entries in RFCBCST.LST and RFCNAMES.LST more carefully, allowing the client to communicate with other systems using TCPBEUI. Or option 151 could be extended to provide a list of many names and/or addresses, a "poor man's" NetBIOS name server.
- Comparing the current date and time with the file date and time in STARTUP.CMD is not the most elegant method of communication between MYOFFICE.CMD and STARTUP.CMD. You should consider using REXX queues, for example, if you need to have the DHCP option handler alert another one of your batch files that an event (receipt of the DHCP option) has occurred. Remember, however, that MYOFFICE.CMD runs for each option and for each lease renewal, so you should be careful not to keep adding to a REXX queue indefinitely to avoid running out of memory. The best approach might be to use a separate queue for each notification and to clean out each queue before use. (See 7.2.3, "Netscape "Message of the Day" Service" on page 295, for an example.)

- Error handling could be substantially improved in these batch files, to catch disk full errors, for example.
- We used the fully qualified domain name merlot.armonk.cooking.net in option 151. We discovered that enabling the OS/2 Warp SOCKS capability in the TCP/IP stack (SOCKS_FLAG "on" in the \MPTN\ETC\SOCKS.ENV file) disrupted name resolution with the fully qualified name if the SOCKS server was unreachable. You may wish to use the hostname only ("merlot" in this example) if you are also using DHCP to pass the standard domain name option and if the hostname alone can be resolved.
- In these batch files there's no guarantee that option 151 handling will be complete before the user attempts a logon. STARTUP.CMD could be enhanced to also check that RFCBCST.LST has been updated and RFCADDR has been run before starting LOGON. Again, REXX queues may help.
- Be aware that using the graphical setup programs in OS/2 Warp (such as the TCP/IP Configuration Notebook or the Multiprotocol Transport Services Setup) can rewrite IBMLAN.INI, DHCP.D.CFG, and other critical files. Make sure that any changes you make to these files in order to support enhanced DHCP option handling are not reversed or modified when you take advantage of the graphical setup programs.
- We do not necessarily recommend placing all the REXX programming code for enhanced DHCP option handling on every client. Instead, you may wish to pass a single DHCP option to a client containing the latest version of the REXX script(s) to be run. This "bootstrap" method can help make managing clients easier.
- Handling TCP/IP printers is also quite easy, and some example REXX scripting appears in DHCPIBM.CMD for the relevant DHCP options (9 and 200).

7.2.3 Netscape "Message of the Day" Service

Based on the file/print example, it should also be possible to provide a "message of the day" service to roaming clients and improve upon the methods used previously. The DHCP server will pass a Web address (URL) to the client as one of the options, and the client will start Netscape at startup to display this message.

7.2.3.1 DHCP Server Setup

Appendix E, "DHCP Options (RFC 2132)" on page 413, lists the DHCP options and their purposes. While not officially part of the standard, DHCP Option 114 is listed as "Universal Resource Locator (URL)," and it seems to

be the most appropriate to pass a Web address to a client. Or, if you prefer, you may use one of the DHCP options strictly reserved for your own use, options 128 through 254.

We added Option 114 with a value of:

```
http://merlot.armonk.cooking.net/siteinfo.html
```

to the same class and subnet as shown in 7.2.2.2, "DHCP Server Setup" on page 286.

7.2.3.2 Web Server Setup

We opted to combine our DHCP server with our Lotus Domino Web server on the same system, merlot.armonk.cooking.net. However, you may wish to have one central Web server with all the "message of the day" Web pages available. For example, you might arrange your Web server to provide access to the following URLs:

```
http://www.siteinfo.megahuge.com/chicago
http://www.siteinfo.megahuge.com/newyork
http://www.siteinfo.megahuge.com/london
http://www.siteinfo.megahuge.com/peoria
```

depending on the location (class, subnet, and so forth) of the client. Assuming there's enough capacity, managing one central Web server can be easier.

However, in this simple example, our DHCP server doubles as our Web server. We placed a test Web page (siteinfo.html) in the root directory of this Web server and verified that the Web page could be loaded across the TCP/IP network using a standard Web browser.

7.2.3.3 Client Setup

Again, in this example, our OS/2 Warp 4 client correctly obtains its DHCP address assignment information from merlot.armonk.cooking.net. Also, Netscape for OS/2 Warp can properly load siteinfo.html from merlot. Now it's up to the client to automatically display that Web page at startup, immediately after receiving the DHCP option.

For this example, you should edit DHCP.D.CFG and add the following line:

```
option 114 exec "$mywebmsg.cmd %s"      # Message of the day (URL)
```

MYWEBMSG.CMD is a REXX script located in some directory in the PATH. Unlike MYOFFICE.CMD in the previous example, which handled two DHCP

options, MYWEBMSG.CMD just handles the contents of the single DHCP option (114) passed to the client.

The "\$" means to wait 30 seconds before starting the REXX script to allow enough time for REXXINIT.DLL to start. REXXINIT.DLL allows REXX scripts to properly function. Alternatively, a "&" symbol can be used to specify a program that requires OS/2 Warp's Presentation Manager. Without the "\$" or "&" symbols, OS/2 Warp's DHCP client software will attempt to run the program immediately upon receiving the option at bootup, possibly before the desktop is ready.

We made some significant improvements to the DHCP option handler, MYWEBMSG.CMD, in order to demonstrate a better way to handle delivery of DHCP options dynamically. MYWEBMSG.CMD obtains the Web address and places it on a REXX queue called MYWEBMSG instead of recording it in a file.

Figure 217 on page 298 displays MYWEBMSG.CMD.

```

/* MYWEBMSG.CMD
 * by Timothy Sipples
 * MYWEBMSG <option data>
 * Return values: 0=successful, 1=input error, -1=system error
 * Designed to handle DHCP option 114 and to place the
 * contents of the option (web address) into a REXX queue.
 */
return_code = 0
q_name = "MYWEBMSG"

/* Get option data from the command line */
parse arg option_data

if option_data = '' then do
    return_code = 1    /* No data returned */
    exit return_code
end

/* Strip leading and trailing spaces */
option_data=strip(option_data)

data=hex2ascii_string(option_data)

if data = '' then do
    return_code = 1    /* String is blank */
    exit return_code
end

/* Put data on REXX queue */
call RXQUEUE "Delete",q_name
call RXQUEUE "Create",q_name
call RXQUEUE "Set",q_name
push data
exit return_code

/* hex2ascii_string (option_data)
 * Take a hex string in the form "nn nn nn ...", verify it's a valid
 * hex string, and convert it to an ASCII character string.
 * Returns the string if the data is valid, a null string if not.
hex2ascii_string: procedure
    parse arg 'hex' ' "' data ' "'
    if (\ DATATYPE(data, X)) then return ""
    data = strip(data,'B','09'x)
    return X2C(data)

```

Figure 217. [OS/2 Warp] MYWEBMSG.CMD

Note that MYWEBMSG.CMD deletes the REXX queue each time it runs, with each lease renewal.

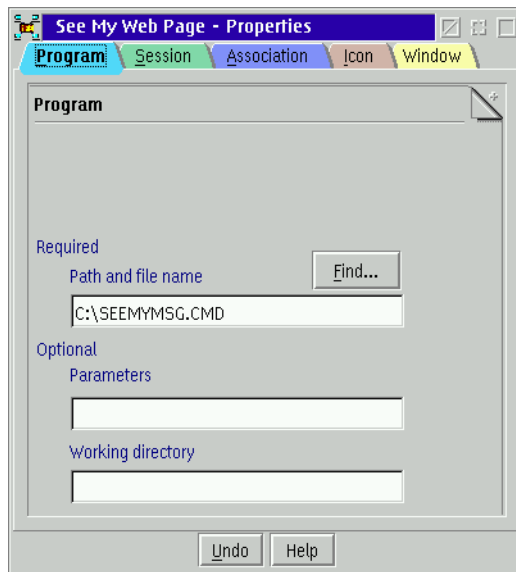


Figure 218. [OS/2 Warp] Program Object in Startup Folder (Page 1)

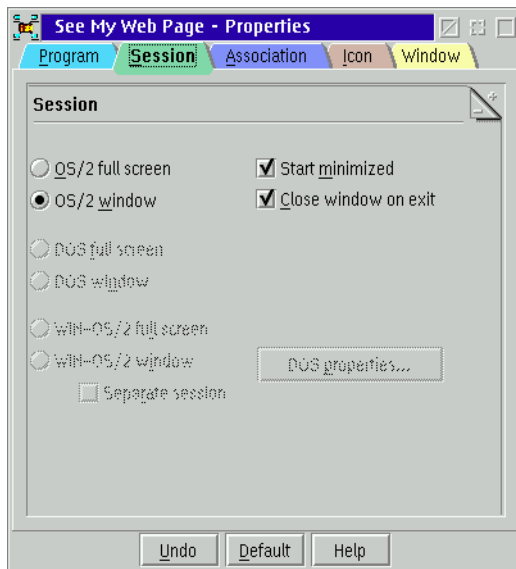


Figure 219. [OS/2 Warp] Program Object in Startup Folder (Page 2)

The companion REXX script, SEEMYMSG.CMD, starts from a program object in the Startup folder. (See Figure 218 and Figure 219 on page 299.) SEEMYMSG.CMD waits forever until the Web address appears in the REXX queue. When it appears, SEEMYMSG.CMD starts Netscape in "kiosk" mode (-k option). In our tests, the "message of the day" appeared approximately 30 seconds after bootup.

Figure 220 displays SEEMYMSG.CMD.

```
/* SEEMYMSG.CMD
 * by Timothy Sipples
 * Waits for web address to be placed on queue MYWEBMSG,
 * reads it, then starts Netscape with the web address.
 * Designed to launch from Startup folder.
 */

q_name = "MYWEBMSG"

address CMD

/* add REXXUtil functions */
call RxFuncAdd 'SysSleep', 'RexxUtil', 'SysSleep'

do forever
  call RXQUEUE "Set", q_name
  if QUEUED() > 0 then do
    parse pull data
    "@START /N /PM NETSCAPE -3 -k "data
    exit
  end
  call SysSleep 5
end
```

Figure 220. [OS/2 Warp] SEEMYMSG.CMD

We prefer the REXX queue method of communication between the DHCP option handler and other REXX scripts. However, both methods can be combined when it's necessary to update a configuration file on the client.

Of course, our simple "message of the day" (Figure 221 on page 301) can be extended to include links to important company Web applications and forms, information on support staff at a particular site, the current stock price, a map of the building, or any other pertinent information that can help a traveler feel at home in a new location. Kiosk mode prevents access to menu and URL controls so that your message (and links) are the only ones available.

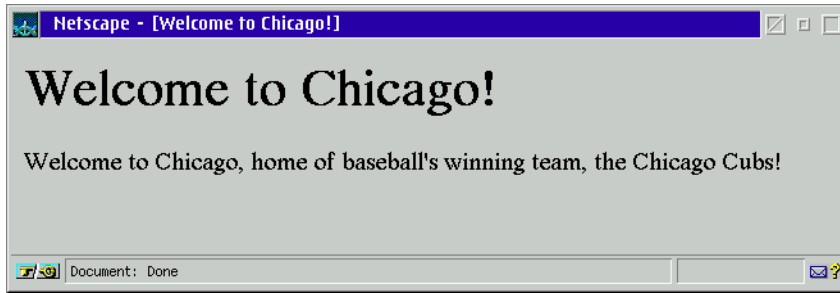


Figure 221. [OS/2 Warp] Netscape in Kiosk Mode

7.2.3.4 Technical Caveats

We recognize that even MYWEBMSG.CMD and SEEMYMSG.CMD are not perfect. Here are some technical issues that you may wish to explore further:

- SEEMYMSG.CMD runs forever (do..forever loop). You may wish to modify this REXX script so that it only tries to obtain the Web address for a particular length of time, perhaps displaying an error message after the time has elapsed.
- If Netscape has been configured with a proxy server of some kind, to access Web pages outside a firewall, then that server should be available from every site. Proxy error messages will prevent Netscape from loading and displaying the "message of the day." If necessary you can use the `-i` command line option to force Netscape to use an alternative NETSCAPE.INI file, one which has proxy capability turned off. That INI file may come from a file server. For example:

```
NETSCAPE -iP:\COMMON\COMMON.INI -3 -k http://www.megahuge.com/rome
```

See the Netscape README file and 7.2.4, "Custom Netscape INI Files" on page 302 for details.

Of course, you can also use Netscape Navigator as a "gateway" to other applications by specifying custom Helper properties in your COMMON.INI file, so that Java and EXE (binary) applications start automatically or start with the selection of a menu option. In other words, you can create a complete desktop environment and use Netscape to deliver it on demand.

- As before, the SOCKS server must be reachable from any site (SOCKS.ENV file); otherwise OS/2 Warp 4 will have trouble resolving fully qualified names.
- Although MYWEBMSG.CMD runs with each lease renewal, there should be little or no performance impact since it's not necessary to rewrite a file.

Also, SEEMYMSG.CMD exits as soon as any Web address appears in the REXX queue.

- Both REXX scripts ought to include more error checking, particularly in the handling of the REXX queue.

7.2.4 Custom Netscape INI Files

While it's certainly possible to pass options to Netscape using the command line, it may also be useful to directly access the NETSCAPE.INI file in order to modify it. Netscape's INI file contains many more options which control the browser's behavior, and it's possible to influence that behavior based on option values delivered by a DHCP server.

For example, roaming users may need to use different proxy or SOCKS firewalls depending on which office (or subnet) they happen to be at. Or your network may be subject to change from time-to-time, with addresses for firewalls changing periodically as your needs change. Changing Netscape settings (particularly complex firewall options) can be tedious for every user; so it makes sense to make these changes automatically and dynamically from a central location.

In this example, we illustrate how a DHCP server can provide the address of a SOCKS server to a client. A DHCP option handler, NSSOCKS.CMD, receives this address and records it in the client's NETSCAPE.INI file. Similar option handlers can be used to modify proxy firewall addresses and other Netscape settings.

7.2.4.1 DHCP Server Setup

By convention, IBM reserves DHCP Option 205 to provide the address of a SOCKS server to a client. (See Appendix E.4, "Unofficial DHCP Options" on page 437.) Therefore, at your DHCP server, you should add option 205 and its value to each subnet, class, or other part of the network where you wish this address to take effect.

For instance, in your network you may have three firewalls:

```
socks1.cooking.net  
socks2.cooking.net  
socks3.cooking.net
```

and each firewall could serve a specific subnet (or group of subnets). Therefore, users on a particular subnet should get the address of the matching SOCKS server. In addition, if one firewall fails, users on the affected subnet could be reassigned to one of the other firewalls by making a simple administrative change at the DHCP server.

Please refer to 7.2.2.2, “DHCP Server Setup” on page 286, for information on how to add a custom DHCP option value to the server.

7.2.4.2 Client Setup

For this example, we again have a fully functioning OS/2 Warp 4 client with Netscape Navigator 2.02 for OS/2 Warp. The client can properly receive regular DHCP address assignment information and can connect to the network.

By default, Netscape Navigator stores its settings in the file NETSCAPE.INI, normally located in the same directory containing the other Netscape program files, such as NETSCAPE.EXE. NETSCAPE.INI may be hidden (or may have the system attribute set). You can remove these attributes by using the command:

```
ATTRIB -R -H -S C:\NETSCAPE\NETSCAPE.INI
```

assuming that you have installed Netscape in the C:\NETSCAPE directory. You may wish to change these file attributes for convenience before working with NETSCAPE.INI. You may also wish to make a backup copy of NETSCAPE.INI before attempting to make changes to it.

In 7.2.3.3, “Client Setup” on page 296, we showed how to install a custom DHCP option handler. For this example, we again installed an option handler on the client, this time NSSOCKS.CMD for option 205. Add the following line to DHCP.DCF:

```
option 205 exec "$nssocks.cmd %s"      #SOCKS firewall server address
```

and place the following REXX script, NSSOCKS.CMD, somewhere in the client’s PATH (see Figure 222 and Figure 223 on page 305):

```

/* NSSOCKS.CMD
 * by Timothy Sipples
 * Example DHCP option handler which changes SOCKS server setting
 * for NETSCAPE.INI file to value specified as an argument (option 205).
 * Return values: 0=successful, 1=error */

return_code = 0
address CMD

call RxFuncAdd "SysLoadFuncs","RexxUtil","SysLoadFuncs"
call SysLoadFuncs

/* Find full path to NETSCAPE.INI */
IniFile = SysSearchPath("PATH","NETSCAPE.INI")
if IniFile = '' then do
    return_code = 1
    exit return_code
end

/* Get option data from the command line */
parse arg option_data

if option_data = '' then do
    return_code = 1
    exit return_code
end

/* Strip leading and trailing spaces */
option_data = strip(option_data)

data = hex2ascii_string(option_data)

if data = '' then do
    return_code = 1
    exit return_code
end

/* Modify NETSCAPE.INI with SOCKS server */
result = SysIni(IniFile,"Services","SOCKS_Server",data)
if result \= '' then do
    return_code = 1
    exit return_code
end

```

Figure 222. [OS/2 Warp] NSSOCKS.CMD (Part 1 of 2)

```

result = SysIni(IniFile,"ProxyInformation","ProxyPref","ManualProxy")
if result \= '' then do
    return_code = 1
    exit return_code
end

/* Set SOCKS server port to 1080 (assumed).
 * Note that 1080 (decimal) is 0438 (hex), which must be in reverse
 * byte order and padded with zeroes (8 digits) for the SysIni
 * function, as shown here. */
result = SysIni(IniFile,"Services","SOCKS_ServerPort",x2c('38040000'))
if result \= '' then do
    return_code = 1
    exit return_code
end

exit return_code

/* hex2ascii_string (option_data)
 * Take a hex string in the form "nn nn nn ...", verify it's a valid
 * hex string, and convert it to an ASCII character string.
 * Returns the string if the data is valid, a null string if not.
 */
hex2ascii_string: procedure
    parse arg 'hex' '' data ''
    if (\ DATATYPE(data, X)) then return ''
    data = strip(data,'B','09'x)
    return X2C(data)

```

Figure 223. [OS/2 Warp] NSSOCKS.CMD (Part 2 of 2)

NSSOCKS.CMD first searches the PATH to locate NETSCAPE.INI using the SysSearchPath function. Then the DHCP option value gets converted from its hexadecimal representation to a text string (such as "socks2.cooking.net"). Finally, NSSOCKS.CMD makes three changes to NETSCAPE.INI using the SysIni function.

7.2.4.3 Other Netscape INI Settings

You can explore NETSCAPE.INI (or any other standard binary INI file used to store application settings) by running REGEDIT2. REGEDIT2 ships with recent IBM FixPaks for OS/2 Warp and OS/2 Warp Server.

Note that the format of NETSCAPE.INI could change; so you should verify that you are changing the correct settings if you upgrade your version of Netscape Navigator.

Here's a partial list of the Netscape settings you may wish to change using DHCP option handlers. To change an INI file setting with REXX, use the SysIni function:

```
result = SysIni(IniFile,Application,Key,Value)
```

NETSCAPE.INI Settings (Partial List)

Application: ProxyInformation

Key: ProxyPref	Value: NoProxy ManualProxy AutoProxy
Key: AutoProxyUrl	Value: (URL string)
Key: FTP_Proxy	Value: (URL string)
Key: FTP_ProxyPort	Value: (8X)
Key: Gopher_Proxy	Value: (URL string)
Key: Gopher_ProxyPort	Value: (8X)
Key: HTTPS_Proxy	Value: (URL string)
Key: HTTPS_ProxyPort	Value: (8X)
Key: Http_Proxy	Value: (URL string)
Key: Http_ProxyPort	Value: (8X)
Key: No_Proxy	Value: (string)
Key: Wais_Proxy	Value: (URL string)
Key: Wais_ProxyPort	Value: (8X)

Application: Services

Key: SOCKS_Server	Value: (URL string)
Key: SOCKS_ServerPort	Value: (8X)

8X = eight digit hexadecimal, reverse byte order

7.2.4.4 Technical Caveats

NSSOCKS.CMD can be enhanced or extended in various ways:

- If you start Netscape Navigator before NSSOCKS.CMD completes its changes to NETSCAPE.INI, then you could pick up old settings. Try adding a REXX queue signalling method to your own DHCP option handler like the one shown in 7.2.3.3, "Client Setup" on page 296. Also, once running, Netscape Navigator may not pick up subsequent changes to NETSCAPE.INI which may occur with each lease renewal.
- This approach can be used with WorkSpace On-Demand, assuming your DHCP option handler has write permission to the appropriate NETSCAPE.INI file(s). Although NETSCAPE.INI files are stored on the file server, and although they can then be centrally managed, you may still prefer to manage certain Netscape settings through your DHCP server

instead. (In fact, the DHCP option handler could run on the file server, in the background, updating every user's NETSCAPE.INI file as needed.)

7.2.5 Summary

By combining custom DHCP options, and providing sophisticated client handling of those options, you can help provide a friendly desktop for your roaming users no matter where they happen to be. Consider using DHCP not just to provide a basic network connection but also to enrich the desktop experience and to allow people to get to work more quickly and easily.

Although we've shown one specific type of client working with these custom DHCP options, you may be able to adapt these techniques for other clients, assuming they support custom DHCP options. Having a scripting language available, such as REXX or BASIC, can help make the task that much easier.

Chapter 8. Security of DHCP and Dynamic DNS

Security is often an afterthought in building a network, but you should be very concerned that your private (or even public) TCP/IP network can prevent unwanted access. In this chapter, we'll show how to refuse connections to the network so that no one with a notebook computer can walk in and grab your information (unless you want them to). One basic security issue is to make sure that you have the latest upgrades to your systems that are depended upon to provide network connectivity. All systems should be at latest upgrades, but especially, those that connect to the network. Manufacturers generally respond fairly quickly when crackers discover another hole in their software.

A good mailing list to be on for security issues is the Computer Emergency Response Team (CERT) list. You get notified of security holes as they are published. See <http://www.cert.org> to get on the mailing list, or <http://www.cert.org/advisories> for the latest information.

8.1 Security Trade-off

For users who want easy access and less procedure to hook up their workstations, DHCP is great and very convenient for both users and the administrator. TCP/IP administrators will be relieved from the tedious work in assigning the IP address to a new request.

For a static IP address assignment, the administrator normally assigns the hostname to the address. It will be treated like a memo. The hostname would be *steven-king*, for example. The assigned hostname is a kind of link to another document, which will have the actual name of the owner and phone number, and so on. However, the whole static TCP/IP itself has no security to prevent an unauthorized access to the IP network. By checking the existing workstation's configuration, people can find the unused IP addresses and required IP router address, then hook up their laptop, for example. At least, we can say it is not so easy to do all this work quickly.

When a DHCP server is up and running, it provides a convenient environment for mobile users. Users can hook up their laptops and instantly get the access to corporate network. In general, DHCP users are anonymous to the administrator. In a company where the basic TCP/IP security guard is required, the basic DHCP environment is not good.

That is probably the one reason why large customers are not eager to use DHCP servers. To breakthrough this problem, we must have another

mechanism in addition to DHCP. Presecured DDNS client and domain design is the answer to the security problem. The TCP/IP administrator might have slightly more workload than the static IP, but the result would be a well protected dynamic IP network that is more secure.

8.2 RSA Public Key Authentication System

Dynamic DNS, as defined in RFC 2137, uses an RSA public key/private key authentication system to secure the dynamic DNS update. A client has a set of keys stored on the hard disk. The key pair is dynamically generated by the client program, NSUPDATE.EXE, or is generated by the administrator through the DDNS Administrator GUI called DDNS Server Administrator. Figure 224 illustrates how the DNS dynamic update is protected with the digital signature authentication.

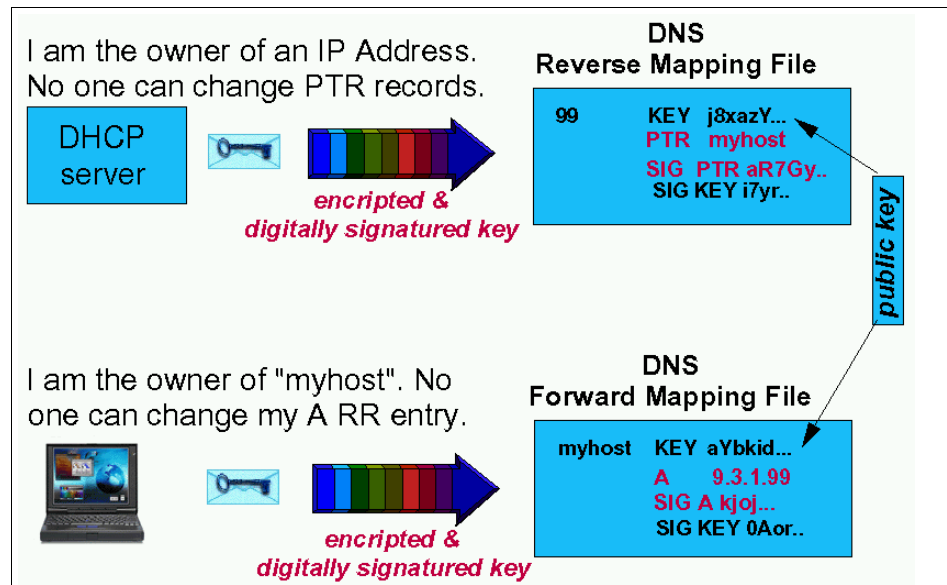


Figure 224. RSA Public Key Authentication System

A client owns the hostname; so it authenticates the A RR (Address Resource Record) in the DNS domain file. If the dynamic zone is created as a Dynamic Secured one, clients can dynamically register A RR at any time. That means the client generates a RSA key pair once at startup time and sends a registration request to a DNS server together with its public key (encoded) and its IP address. The important matter is that the client keeps the private key on a hard disk or a networked drive, and it should not be exposed.

The RSA key system depends on the fact that the digital signature can be verified with the client's public key, and the private key cannot be discovered from either the public key, or the digital signature, or a combination of both. The client must generate a new digital signature each time an update request is sent to DNS.

Once a hostname is registered with the public key and the digital signature, the entry is verified or authenticated at each DHCP renewal time or when the DNS TTL (Time-to-Live) is expired.

If the dynamic zone is defined as a Dynamic Secured one, clients cannot add or update their hostname dynamically. Since the administrator defines hostnames and generates key pairs for each hostname, the client should imports the specific key pair and must use it for the further processing.

The illustration of Figure 224 on page 310 simplifies the record entries of both domain file (forward mapping file) and reverse file (reverse mapping file), but the basic is there are two SIG records and one SIG record that authenticate the KEY record. Another SIG record authenticates either the A record or PTR record. The part of actual domain file is shown in Figure 225.

```

client1 4660 IN HINFO "Ni0weDAwMDBlNTY4N2Y2MA==" "IBMDNS-PROXY" ;Cl=3
4660 IN SIG A 1 4 4660 2147483647 893107703 0x6d25
      client1.austin.cooking.net u1SD2P.....G5VK47mig== 2147483647 ;Cl=3
4660 IN SIG HINFO 1 4 4660 2147483647 893107703 0x6d25
      client1.austin.cooking.net HrwTz.....e4p+a/g== 2147483647 ;Cl=3
4660 IN SIG KEY 1 4 4660 2147483647 893107703 0x6d25
      client1.austin.cooking.net FFqsEa.....Yujelkw== 2147483647 ;Cl=3
3600 IN KEY 0x0000 0 1 AQPqg+gV....VuGWzJW3X ;Cl=3
4660 IN A 192.168.7.22 ;Cl=3

```

Figure 225. [Warp Server] Example of KEY, SIG and A Resource Record

The highlighted field in a KEY record is an encoded public key of the client. Due to space limitations, the whole key is not shown here. The actual key length is 1024 bits (128 bytes), but it is encoded to 88 bytes. A digital signature is also 1024 bits but again encoded to 88 bytes.

You probably heard in the news that the U.S. Government approved the 1024-bit RSA key system for export, since it is only used for authentication and not to encrypt messages. For the purpose of message encryption, the key length is restricted to 56 bits for the export version and 128 bits for U.S. domestic version only.

We distinguish two different dynamic DNS zones:

- Dynamic Secured

The Dynamic Secured zone provides a great deal a secured zone. Once a client's hostname is dynamically registered in the DDNS database, it is protected and cannot be taken over by another client. The workstation with the DDNS client function can dynamically join the network and can register its name, and because of that, the DNS zone is open for all clients.

The Dynamic Secured zone buys you tremendous flexibility, given the goal of an administratorless system, in that it requires no administrator intervention to have the DHCP clients get registered, which exactly is the point of DHCP. The only shortcoming of this concept is that a malicious client could create useless names, but then, this shouldn't bother.

However, be aware of the fact that any DDNS client could perform the very first registration and then all of a sudden would own a hostname that might be rather critical than useless.

- Dynamic Presecured

If you are looking for *the* perfect secured DNS zone you need to look at a so-called Dynamic Presecured zone. Clients joining the presecured domain must have the key pair generated by the administrator. This key then can be distributed to the client through a networked drive, through e-mail, or through a diskette. Using this method, there is no chance for clients to [unlawfully] join the DNS space. The administrator would have a perfect control over their DNS zone.

As for the client's public key, it is not a secret one. It is even intended to be known by everyone. For example, the client's public key can be retrieved through the `NSLOOKUP` interface. Figure 226 on page 313 is an example of the `NSLOOKUP` output which provides all information of `client1.austin.cooking.net`.

```

Default Server: fajita.austin.cooking.net
Address: 192.168.7.10

> > Server: fajita.austin.cooking.net
Address: 192.168.7.10

client1.austin.cooking.netCPU = Ni0weDAwMDBlNTY4N2Y2MA==OS = IBMDDNS-PROXY
client1.austin.cooking.net
Signature Record covering A RR's
Authentication Algorithm = 1 (MD5/RSA) Labels = 4
Original TTL = 4660 (1 hour 17 mins 40 secs)
Client SIG expiration = 2147483647, Mon Jan 18 21:14:07 2038
Time signed = 893107703, Mon Apr 20 16:28:23 1998
Server SIG expiration = 2147483647, Mon Jan 18 21:14:07 2038
Key footprint = 0x6d25
Signer's name = client1.austin.cooking.net
Signature = u1SD2P0kC8Idl0HyelcqqZIRrMEkHQ3e2SdmwaaStBRRcSBC4DDRMSIRDwvgwcrA7wfDvY9
QQEoOKG5VK47mig==
client1.austin.cooking.net
Signature Record covering HINFO RR's
Authentication Algorithm = 1 (MD5/RSA) Labels = 4
Original TTL = 4660 (1 hour 17 mins 40 secs)
Client SIG expiration = 2147483647, Mon Jan 18 21:14:07 2038
Time signed = 893107703, Mon Apr 20 16:28:23 1998
Server SIG expiration = 2147483647, Mon Jan 18 21:14:07 2038
Key footprint = 0x6d25
Signer's name = client1.austin.cooking.net
Signature = HrwTzfLIGkvFfUmHwKr6aaHV/qO6RVc6uzItH9cYYQsld1PRVgGrENlQZscj2gwcY4Z
B/LfBcHlfCVse4p+a/g==
client1.austin.cooking.net
Signature Record covering KEY RR's
Authentication Algorithm = 1 (MD5/RSA) Labels = 4
Original TTL = 4660 (1 hour 17 mins 40 secs)
Client SIG expiration = 2147483647, Mon Jan 18 21:14:07 2038
Time signed = 893107703, Mon Apr 20 16:28:23 1998
Server SIG expiration = 2147483647, Mon Jan 18 21:14:07 2038
Key footprint = 0x6d25
Signer's name = client1.austin.cooking.net
Signature = FFqsEaYujelrDfRxlS77NxE5Om/1+OgvRUN/P2QLPiNIYa5S1FdNbttBi3lOQRRs6eRje
fwjJFBlIqqPVuX8kw==
client1.austin.cooking.net
flags = 0x0000 (HostKey) protocol = 0, algorithm = 1 (MD5/RSA)
public-key data = AQPqq+gVMx4td+vg1qCjv3fkngVyNn++w/uAoNHA/t52qzVkyXOYAGftLS78gi
pWJtVpDGigMWs7ly3VuGWzJW3X
client1.austin.cooking.netinternet address = 192.168.7.22
austin.cooking.netnameserver = fajita.austin.cooking.net
fajita.austin.cooking.netinternet address = 192.168.7.10
>

```

Figure 226. [Warp Server] NSLOOKUP Output Returning Public Key

8.3 Getting More Information from the Client to DNS

With TCP/IP Version 4.1 for OS/2, the OS/2 DDNS client has a new function which can force the client to input more practical information to the DNS database:

- option 192

Option 192 is defined in the DHCP server configuration file, and it enables the DHCP client to load a special window for the user to input predefined data, such as:

- Office and building number
- Name
- Phone number
- E-mail address

Figure 227 on page 314 shows how the administrator defines option 192 at the DHCP server.

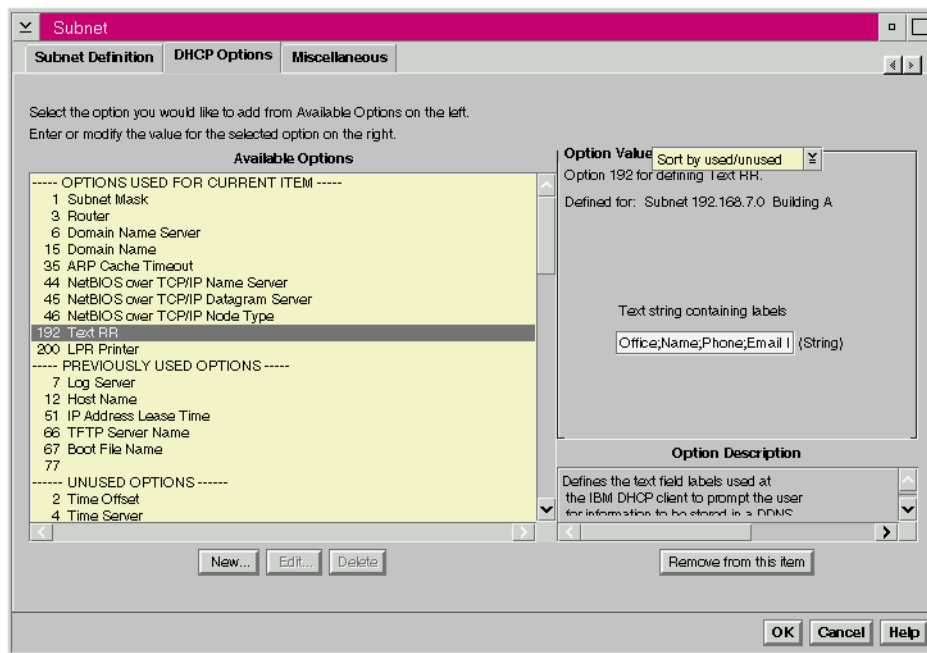
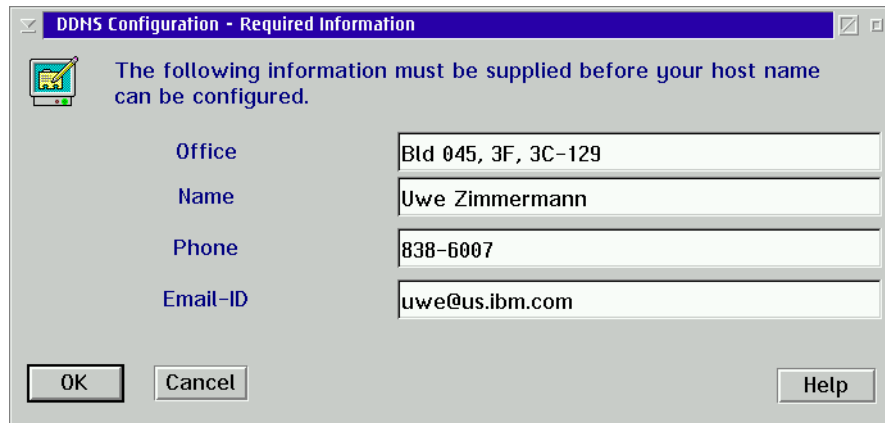


Figure 227. [Warp Server] DHCP Server, Configuring Option 192 TXT Record

To enable option 192, type the required input data in the text string containing tablets field. For example:

Office;Name;Phone;Email-ID;

When the DHCP client gets control from the server, the server initiates the DDNS client to prompt the user to input data, as shown in Figure 228 on page 315.



DDNS Configuration - Required Information

The following information must be supplied before your host name can be configured.

Office	Bld 045, 3F, 3C-129
Name	Uwe Zimmermann
Phone	838-6007
Email-ID	uwe@us.ibm.com

OK Cancel Help

Figure 228. [Warp Server] DDNS Client Prompts for the Input

8.4 Presecured Domain

The presecured domain is the most secured DNS name space available. For information on how to set up a presecured domain, refer to 4.7.6.3, "Dynamic Presecured Mode" on page 130. You can set up a sub-domain in the corporate root domain and make the sub-domain work as a Dynamic Presecured domain using the TCP/IP 4.1 DDNS.

Two reasons for a higher security standard would be:

- The DHCP client cannot register its hostname without having a RSA key pair and the DDNS client code present.
- Without a RSA key pair, the PTR record for reverse mapping cannot be registered either, so that reverse mapping would fail, and the administrator would be able to know which IP addresses are unofficial users. The administrator would be able to determine the MAC address of those DHCP clients' LAN adapters. Then, at another step, the administrator could configure the DHCP server to exclude those unlawful clients through the server configuration GUI.

Just like a Lotus Notes administrator generates an ID file for the user and distributes it to users, the TCP/IP administrator generates a key file and passes it to the client.

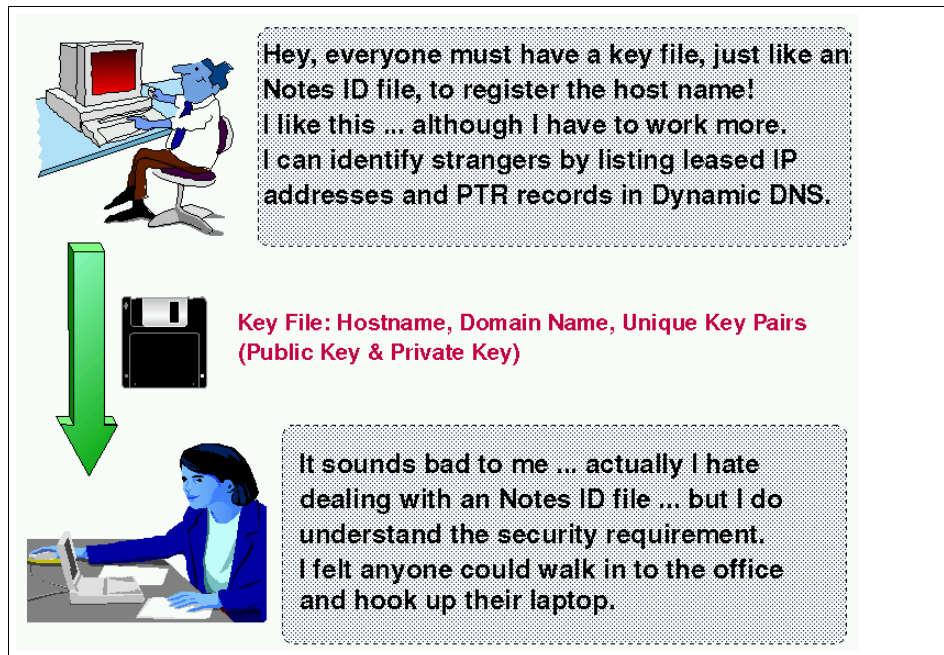


Figure 229. Presecured TCP/IP Domain Concept

Although the TCP/IP administrator would have a higher workload establishing a presecured domain in comparison to working with a static DNS environment, the presecured domain ensures a well-trusted DNS name space, and it can reduce the total cost of administration.

The most reasonable implementation of a presecured domain is to combine it with static DNS servers and create a kind of hybrid domain design, as shown in Figure 230 on page 317.

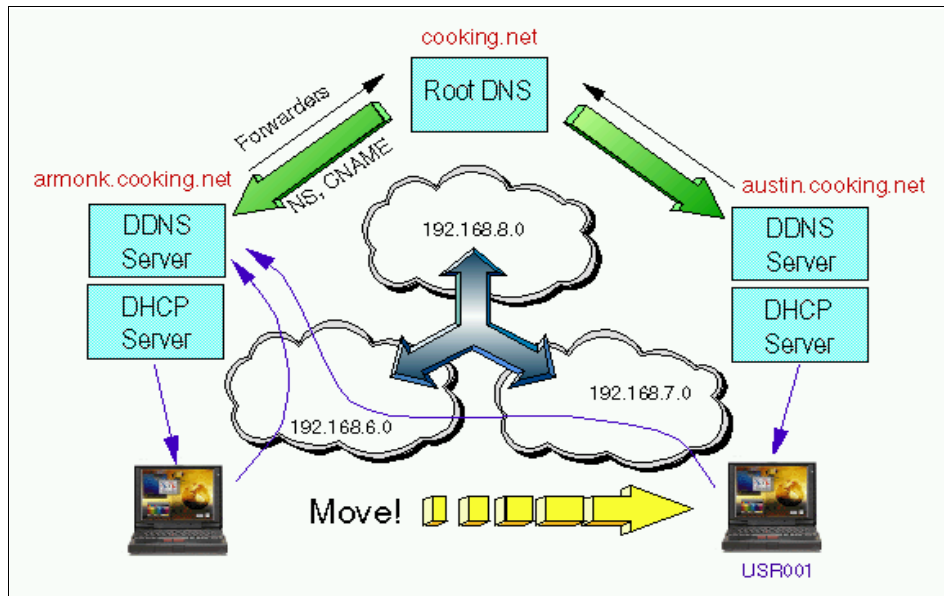


Figure 230. Combination of Static DNS and Dynamic Presecured DNS

This design ensures that an administrator pre-generates hostnames and corresponding key files in advance. Then, the administrator would create a CNAME (alias) record in the root DNS table when a new request for the hostname is received.

The following is the example of CNAME entries in the root DNS table:

uwe-zim	IN	CNAME	usr001.armonk.cooking.net
michaelmd	IN	CNAME	usr002.austin.cooking.net

Let's look at the hostnames from a generic view. The user uwe-zim would always be known in the network as uwe-zim, regardless of what the hostname in the FQDN looks like. You can set up the all DHCP/DDNS servers in your network using this setup scheme. When the client machine is moved to another location, the location's DHCP server will serve the new IP address for it, but the client machine can still update its home DDNS server with the new IP address.

This design allows the safe and smooth deployment of Dynamic DNS together with DHCP servers.

8.5 ProxyArec Consideration

The ProxyArec function is desired by the industry, and several companies provide products that embrace ProxyArec. ProxyArec is based on the IETF Internet draft called *Interaction between DHCP and DNS*. This draft can be found on the Internet at the following Web site:

<http://www.ietf.org/internet-drafts/draft-ietf-dhc-dhcp-dns-08.txt>

The latest draft may have a newer number.

Besides the TCP/IP Version 4.1 DHCP and DDNS servers, we also tested NTS Shadow IPserver which also provides a ProxyArec function. However, NTS Shadow IPserver does not support option 81, which tells the DHCP server that a DHCP/DDNS client wants to register/update its A RR by itself; so the DHCP server would not perform the ProxyArec function to update the DDNS table. Option 81 is discussed in 8.5.1 on page 320.

When the DHCP server is configured to enable the A resource record update in addition to the PTR resource record update, sometimes called a ProxyArec function, any DHCP client which carries the option 12 hostname will register both A RR and PTR RR. Since the DHCP server uses its own RSA key for all DDNS update requests, there is no authentication mechanism to validate that the client really is the client that has the original hostname.

However, the ProxyArec function in TCP/IP Version 4.1 for OS/2 DHCP server provides security. When the DHCP.D.CFG file in the \MPTN\ETC subdirectory has the following statement, the client's MAC address will be used to memorize the ownership of the A RR:

```
ProxyArec Protected
```

To disable the MAC address identification, the statement should be:

```
ProxyArec Standard
```

Then, the DHCP server will blindly update the A RR without comparing the MAC addresses at all.

From the DHCP Server Configuration GUI, the ProxyArec settings require you to complete two panels. First, you need to specify a domain name and a DDNS server IP address (or hostname) in the DHCP Server Parameters window, as shown in Figure 231 on page 319.

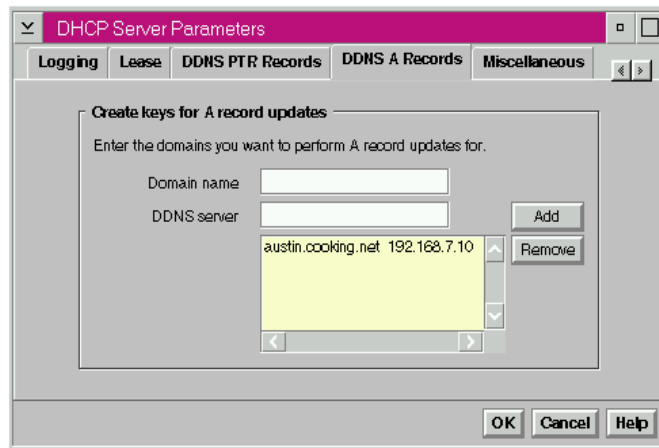


Figure 231. [Warp Server] ProxyArec - Enable DHCP Server to Update A RR

The Global Parameters window, as shown in Figure 232, is the only place where you would enable ProxyArec. You must have a Dynamic DNS server to enable dynamic A RR updates.

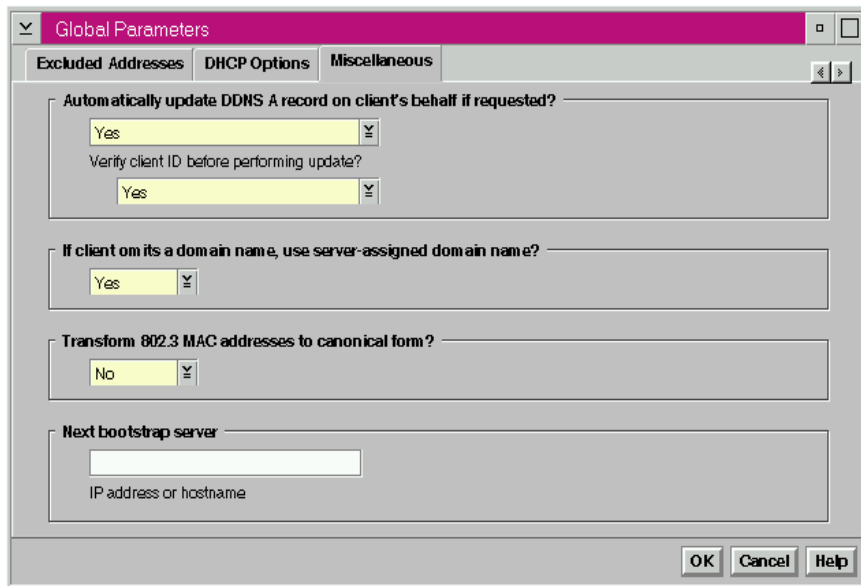


Figure 232. [Warp Server] Enable ProxyArec and Option for Protection

The question in Figure 232 on page 319 that says `Automatically update DDNS A record on client's behalf if requested?` is defaulted to **No**. If you select **Yes** from the pull-down menu, the ProxyArec function is enabled. Another selection box `Verify client ID before performing update?` decides on the two modes of ProxyArec:

- Standard
- Protected

To enable protected ProxyArec, select **Yes** from the pull-down menu.

Remember that the main purpose of the ProxyArec function is to support non-DDNS clients to dynamically register/update its hostnames where security is not a first priority. Configuring the DHCP server to use the client's MAC address to protect unauthorized update from a different client does not indicate security. Dynamic DNS name space is almost like a public space. Any DHCP client can register the hostname with the DHCP option 12.

Windows 95 and NT handle option 12 in a different way than OS/2. Even though the Windows 95/NT client was provided with a hostname in the TCP/IP DNS settings, the hostname is not used. Instead, the Computername definition is used as option 12. This is a unique (proprietary) Microsoft implementation of a hostname. Traditionally, the computername, also known as the NetBIOS name, is not controlled by the IS department or TCP/IP administrators; so there might be a chance that people use the same computername in a TCP/IP network, which usually results in duplicated names. Duplicate names results in not opening LAN adapters.

For your information, WorkSpace On-Demand Release 1 uses the ProxyArec function when DDNS is required. IBM PPP server on top of Warp Server's Remote Access Server also uses the ProxyArec, because it supports non-LAN distance clients, such as Windows 95 or Windows NT Workstation.

8.5.1 ProxyArec and Option 81

We don't recommend having a ProxyArec DHCP server and standard DDNS clients in a same location because a DDNS client cannot use its own interface to the DDNS server. A RR will be owned by the DHCP server. However, in such a situation, the OS/2 DHCP/DDNS client can use a special DHCP option, which is option 81, which would disable the ProxyArec function for a requesting client.

Unfortunately, there is no graphical user interface where you could define option 81. You must manually add option 81 to the

\\MPTN\ETC\DHCPD.CFG file using HEX translated FQDN (Fully Qualified Domain Name). This is shown in Figure 233.

```
# Basic options required

clientid  MAC
interface lan0

# Uncomment as desired for logging
numLogFiles  4
logFileSize  100
logFileName  dhcpd.log
logItem      SYSERR
logItem      OBJERR
logItem      PROTERR
logItem      WARNING

option 12 tuna                                # Host name
option 81 HEX "00 00 00 74 75 6E 61 2E 61 75 73 74 69 6E 2E 63 6F 6F 6B 69 6E 67 2E 6E 65 74"
# Don't ProxyArec tuna.austin.cooking.net

updateDNSA "nsupdate -h%s -d%s -s"d;a;*;a;a;s;s;s;3110400;q" -q"
updateDNSTxt "nsupdate.exe -h%s -s"d;txt;s;a;txt;s;s;s;3110400;q"
```

Figure 233. [OS/2 Warp] DHCPD.CFG Example with Option 81

We wished we could specify an ASCII name rather than HEX code. But it would not work. The IETF document *Interaction between DHCP and DNS* defines option 81, as shown in Figure 234:

Code	Length	Flags	Rcode1	Rcode2	FQDN
81	n	00	00	00	

Figure 234. IETF Definition of Option 81

The Flags indicates the DHCP server to do the following:

- 00 Client wants to be responsible for updating the FQDN to IP address mapping.
- 01 Client wants the server to be responsible for updating the FQDN to IP address mapping.

Rcode1 and Rcode2 define the response code area used by the DHCP server to check a DHCP client's response for Dynamic DNS updates.

8.6 Securing Lease Allocations

There is no one solution to controlling which clients can obtain a DHCP lease from a server, or if they should get a lease at all. However, there are some alternatives available to you:

- You can use classing (option 77) when configuring your address pools. Find an example on classing on page 287.

Using classes requires you to visit each client and modify their configuration. Once you have your classes established, only devices that belong to the class will be allocated a lease.

Be aware that not all clients can make use of option 77.

- At the server, you could allocate IP addresses to individual MAC addresses. This can be very labor intensive, but ensures that only machines with known MAC addresses will obtain a lease from a server.
- You may be able to find DHCP server software that allows you to list which MAC addresses the server will accept. DHCP servers that support roaming machines may be adapted to such use.
- If you want to ensure that clients are only served by a particular server, you could conceivably install one DHCP server per subnet. This is not a good solution, but if you choose to do this, ensure that your routers do not forward broadcast traffic, and that there is no DHCP Relay Agent on the subnet.

This solution does not stop any malicious user from connecting to the network and obtaining an IP address.

8.6.1 Preventing Access to Unauthorized Devices

This would have to be done using a mechanism other than DHCP. DHCP does not prevent other clients from using the addresses it is set to hand out nor can it distinguish between a computer's permanent MAC address and one set by the computer's user. DHCP can impose no restrictions on what IP address can use a particular port nor control the IP address used by any client. You may be able to use classing to limit leases to only those clients that belong to a valid class.

8.6.2 'Rogue' DHCP Servers

It is possible that a malicious or inexperienced user could create problems on your network by setting up an unofficial DHCP server.

The initial problem is that the server could pass out IP addresses already belonging to some other computer. It is possible that you could end up with two or more devices ending up with the same IP address. The end result is problems using the devices. These problems can be intermittent or the devices may fail entirely, necessitating a restart.

Other problems are possible if the unofficial server gets a device to accept its lease offering, and then passes to the device its own (possibly incorrect) DHCP options. For example, if options 1 or 3 are incorrect, the device may not be able to communicate with devices outside its own subnet (or in some cases, with devices on the same subnet)

Another scenario to consider, is a device that loads its operating system over the network using TFTP. If this device is directed to load a different file (possibly on a different server), it allows a perpetrator to take over the client. Given that boot parameters are often made to control many different things about the computers' operation and communication, many other scenarios are just as serious.

Note that by using BOOTP you are exposed to the same vulnerabilities.

8.6.3 Connecting to Untrusted Networks - Firewalls

There may come a time when you want to connect your private intranet to an untrusted network. What is an untrusted network? Well, the biggest example is the global public Internet (maybe it is time to have your own local mail or Web server rather than having your ISP host it for you). Another example may be one of your customers' networks. Say you are a consulting firm who wants to provide remote IS services for your clients. One way to do this is to connect your two corporate networks together. You would not want all your customers, and whomever they have logged onto their networks, to have access to all machines on your private network. You probably want them to have access to your Web server, and not much else. However, you need access to machines on their network so that you can do the appropriate maintenance, upgrades, monitoring, and so forth, to perform your IS duties. There are many more examples of why you might want this kind of protected architecture between your private internet and other untrusted networks. This section will show you the how-to more than the why do it. Be aware that there are more methods to connect than the single one we are showing you.

To interconnect safely, you will need a firewall. A firewall is not a panacea for security, and you should not treat it as such. A firewall can help provide security, and there are many more issues beyond the firewall to keep in mind.

Firewalls provide secure access from a company's internal TCP/IP network to the public Internet. Intruders are blocked and cannot access the internal network from the public network without authorization. Internal users, however, can still access resources on the public Internet. In combination with good security practices, firewalls can help keep private information on your network secure.

Several vendors produce high quality, full-featured firewalls. Examples include IBM Firewall for AIX and Windows NT and LanOptics Guardian for OS/2 Warp and Windows NT. Alternatively, many firms opt to contract for firewall services with their network providers to avoid having to learn the specialized skills required to manage network security.

Most firewalls can support two methods used by internal clients to access public Internet sites. SOCKS is the most modern method, and it can provide transparent access for all TCP/IP-based applications if the client operating system has a SOCKSified protocol stack. OS/2 Warp and WorkSpace On-Demand have built-in SOCKS support; third parties provide SOCKS extensions for Windows 95 and Windows NT clients. Practically all Web browsers support SOCKS-based firewall access.

Proxy servers can also provide access to public sites through a firewall for internal clients. However, proxy servers can only handle HTTP, FTP, and Gopher protocols; so they are mainly of benefit to Web browsers. Caching proxy servers, such as IBM's Web Traffic Express, can even cut down on network traffic through the firewall by keeping frequently accessed Web pages and files on the proxy server itself.

In principle, a proxy server, such as IBM Web Traffic Express or Domino Go Webserver, can be used as a simple firewall solution when some degree of security is needed. A proxy server performing that role on Warp Server, for example, would need two network adapters (one for the internal network and one for the external network) and a proxy server software package installed and running. That server should have an `IPGATE OFF` command in its `\MPTN\BIN\SETUP.CMD` file so that TCP/IP traffic will not be forwarded from one LAN adapter to the other. (Otherwise, an intruder could access internal systems.) Also, it should not have any daemons running which may expose vulnerabilities. Nor should the proxy server have any LAN services bound to the adapter handling external network access. Moreover, the proxy server should not handle any requests from external users. Provided these conditions can be met, a proxy server can act as a simple firewall.

You may wish to download a 60-day evaluation version of IBM Web Traffic Express to learn how to set up a caching proxy server. Please visit:

<http://www.software.ibm.com/webserver/wte/index.htm>

to obtain your own evaluation copy. An evaluation copy of Lotus Domino Go Webserver can be found on the Internet at:

<http://www.ics.raleigh.ibm.com/dominogowebserver>

Naturally, commercial grade firewall products go well beyond simple proxy server functions. For a full description of how to set up a firewall and institute good security management, we recommend consulting a redbook specifically devoted to firewall security, such as *A Comprehensive Guide to Virtual Private Networks, Vol.1: IBM Firewall, Server and Client Solutions*, SG24-5201.

8.6.4 Connecting Through Untrusted Networks - VPN

A virtual private network (VPN) is an extension of an enterprise's private intranet across a public network such as the Internet, creating a secure private connection, essentially through a private tunnel. VPNs securely convey information across the Internet connecting remote users, branch offices, and business partners into an extended corporate network.

Internet Service Providers (ISPs) offer cost-effective access to the Internet (via direct lines or local telephone numbers), enabling companies to eliminate their current, expensive leased lines, long-distance calls, and toll-free telephone numbers.

A 1997 VPN Research Report, by Infonetics Research, Inc., estimates savings from 20% to 47% of wide area network (WAN) costs by replacing leased lines to remote sites with VPNs. And, for remote access VPNs, savings can be 60% to 80% of corporate remote access dial-up costs. Additionally, Internet access is available worldwide where other connectivity alternatives may not be available.

The technology to implement these virtual private networks, however, is just becoming standardized. Some networking vendors today are offering non-standards-based VPN solutions that make it difficult for a company to incorporate all its employees and/or business partners/suppliers into an extended corporate network. However, VPN solutions based on Internet Engineering Task Force (IETF) standards will provide support for the full range of VPN scenarios with more interoperability and expansion capabilities.

The key to maximizing the value of a VPN is the ability for companies to evolve their VPNs as their business needs change and to easily upgrade to future TCP/IP technology. Vendors who support a broad range of hardware

and software VPN products provide the flexibility to meet these requirements. VPN solutions today run mainly in the IPv4 environment, but it is important that they have the capability of being upgraded to IPv6 to remain interoperable with your business partner's and/or supplier's VPN solutions. Perhaps equally critical is the ability to work with a vendor who understands the issues of deploying a VPN. The implementation of a successful VPN involves more than technology. The vendor's networking experience plays heavily into this equation.

8.6.5 TFTP Security

If you decide to provide BOOTP services, remember that TFTP is inherently insecure. On TCP/IP Version 4.1 for OS/2, you can specify which client IP addresses have access to the TFTP directory. For example, our \MPTN\ETC\TFTPAUTH file contained the following lines:

```
C:\TFTPBOOT RO 192.168.6.17
```

```
C:\TMP RO 192.168.6.11
```

which means that the machines with the respective IP addresses have read only access to the directories listed. Since we can preassign IP addresses based on the machines MAC address, this is fairly safe.

Chapter 9. Reliability

"I can't get on the network!"

Those six words should never be heard by a network manager. Yet with TCP/IP networks, they are, unfortunately, all too common.

This chapter focuses on how to make your TCP/IP network super reliable, with connections always available and conflicts eliminated. In particular, we'll explain how to create "fault tolerant" TCP/IP address servers.

If your TCP/IP network does not need to be reliable, please skip this chapter.

9.1 Battlefield Questions

To design a network for reliability, you might think of your network as a battlefield. The goal is to assure that your messenger can travel from one end of the battlefield to another without getting blown up. In between are roads, bridges, highways, airways, rail lines, and other mechanisms used by your soldiers to communicate. If one (or more) of these conveyances gets blown up, your courier must still reach their destination.

In designing your network, you should ask yourself a series of "What if?" questions. What if the router shorts out? What if the DHCP server is buried by a mudslide? What if a remote office satellite link gets blocked because of sunspots?

Then, when you have examined the "What if?" questions, there are some more questions to answer:

- How likely are these events?
- How much reliability do I need? For whom?
- Where can I spend limited dollars to get the highest possible degree of reliability?
- If a failure does occur, will I know about it? How fast can I respond?
- Do I regularly rehearse how to respond to failures?
- Do I have a written plan of action to handle network outages?
- Am I measuring the frequency and severity of network problems so I can determine whether service improves?

If you're doing your job, these questions never have permanent answers. You should always regularly reexamine these issues, even if your network runs smoothly.

9.2 Failure Events

Datagrams can only travel across your network as long as every part of the link works, from end-to-end. Your network can only be as reliable as the weakest link.

9.2.1 Severed Connections

One obvious possible failure is a severed connection. Wires may be cut, cables inadvertently unplugged, or fiber optics dug up by a backhoe. TCP/IP can easily reroute around such failures provided another connection is available. An easy way to design redundancy into connections is to simply provide two (or more) separate lines or wires, preferably using different technologies. Another way is to structure the network as one unbroken ring (or circle). If one connection in the ring does break, traffic can still flow throughout the network because all points are still connected.

The U.S. Federal Aviation Administration, for example, discovered that their connection redundancy failed, and air traffic control services were disrupted for a time. (Fortunately, no one suffered any injury.) The FAA uses telephone lines to connect various facilities. These facilities each have multiple circuits, and switching equipment can quickly reroute network traffic over backup circuits. However, the FAA granted a single long distance telephone company the contract to connect their facilities. When the telephone company's own network failed, the FAA had no backup, and the multiple circuits didn't matter. Now, the FAA has contracted with a second long distance telephone company to provide backup connections should they be needed.

In another example, PanAmSat's Galaxy IV satellite suffered a catastrophic failure in 1998. Amazingly, almost none of the U.S. paging services had any immediate backup satellite, so paging services were disrupted for days. Physicians, in particular, were hard hit, because they lost contact with hospitals and could not be called in by pager.

9.2.2 Facility Loss

Fires, floods, earthquakes, hurricanes, power outages, and other catastrophes can wreak havoc with a network by knocking out crucial servers, routers, bridges, switches, and other devices. Large parts of the

Internet, for example, were disabled when the last San Francisco earthquake knocked down some key buildings.

Many companies do not properly plan for these contingencies. Your network may be required to handle these events, so try to avoid placing all your vital network devices in one location.

An uninterruptable power supply (UPS), with a remote alerting mechanism, can help keep critical network systems up-and-running. However, backup electrical power should be provided for all critical systems. It's easy to forget that routers, bridges, concentrators, hubs, servers, switches, and even clients all need backup power if they are to continue operating.

9.2.3 Router Outages

Subnets in a TCP/IP network can quickly become cut off from the rest of a network if a router fails. These devices should be extremely reliable, and many network designers choose dedicated pieces of industrial equipment from IBM, Bay Networks, Cisco, and other vendors instead of less reliable PCs. Routers can also be installed in pairs.

9.2.4 DHCP Server Problems

DHCP can help provide added flexibility and convenience in managing a TCP/IP network. Static addresses can often conflict, locking systems out of a network. Also, static configurations can only be changed with a significant amount of work, making it difficult to deal with router outages, changes in subnet structures, and so forth. In other words, DHCP servers can actually help ensure reliability.

However, if a DHCP server fails, then a TCP/IP network can quickly run into trouble. DHCP servers must be reliable. (See 9.3, "A "Fault Tolerant" DHCP Server" on page 331, for one way to ensure DHCP server reliability.)

Failures of DHCP relay agents can also prevent new systems from joining a TCP/IP network. We recommend using DHCP relay agents sparingly. Instead you may wish to have dedicated routers handle the task with BOOTP relays as needed. If you do use DHCP relay agents consider a more fault tolerant solution, such as the one described later in this chapter.

9.2.5 Name Server Difficulties

Without name servers, your users will have to resort to numeric IP addresses, effectively making your network useful only to the most savvy technicians. DNS servers must also be active and reachable at all times.

TCP/IP clients, including OS/2 Warp 4 and Windows 95, can readily accept two or more name server addresses. If the first name server can't be reached, the client will attempt to contact the backup name server(s). We recommend taking advantage of this simple feature and to use two (or more) name servers on your mission critical network.

Dynamic DNS servers, such as OS/2 Warp Server, also help minimize network problems. For example, if an important Web server fails, its name can be reassigned to another numeric IP address that much more quickly and easily with a DDNS server. In fact, this secure update may be triggered automatically when a server failure has been detected by network alerting tools.

9.2.6 Other Server Vulnerabilities

Server failures can prevent users from accessing important applications and databases. Technologies, such as RAID, can help cope with hard disk failures. Many vendors provide fault tolerant server solutions, including IBM. Storage management software, such as IBM's ADSM, help send backup copies of files across the network to off-site locations.

Yet with all these server technologies, it's important to remember that most failures occur because of software problems. All the RAID storage in the world won't help if your server suffers from the blue screen of death. Some operating systems can be vulnerable to failure, especially when intentionally induced. For example, in the late 1980s, Robert Morris at Cornell University discovered a security vulnerability in some versions of UNIX. He used this vulnerability to spread a worm which quickly infected numerous systems connected to the Internet. For several hours, the Internet essentially ground to a halt as system managers attempted to address the problem. Many could not even log onto their own servers since the worm multiplied rapidly, consuming practically all the attention of the server's processor(s). More recently, some versions of Windows NT have been vulnerable to attack using ping and/or Telnet.

You should keep close watch on your servers and regularly review software bug reports and other information in case others find additional problems which may affect you. Antivirus software, such as Norton Antivirus, may detect and eradicate PC viruses on your servers. (Viruses can also cause network problems.)

9.2.7 Client Failures

Many network problems really relate to the fragility of traditional PC clients, which tend to break down all too often. It isn't hard to delete an important icon, modify a critical system file, or otherwise render a traditional PC unable to connect to the network. PC users then have to wait for time consuming repairs to software, hardware, or both.

Various estimates peg the cost of managing a typical business PC (the annual total cost of ownership) at between \$5,000 and \$26,000. These costs appear to be rising despite (or perhaps because of) advances in technology. While that annual cost may be perfectly justified for many users, including programmers, engineers, heavy office suite users, hobbyists, and others, the high cost is not necessarily attractive to everyone. Yet, the benefits of PC applications, graphical user interfaces, and online help do make sense.

Many companies in the computing industry recognize these problems and are working to solve them through network computing. True network computing solutions provide most or all of the benefits of PC applications with a total cost of ownership approaching that of a mainframe or UNIX terminal. Examples include network computers from IBM, Sun, Oracle, and others, Java, and IBM's Workspace On-Demand for Intel-compatible systems. One of the key benefits is that users can log on from any station on the network and get access to individual documents, applications, and Web pages from anywhere. Consequently, network computing can improve the reliability of the whole network by providing access to it more often and with less complexity.

While some users do need the mainframe on a desk that the PC has become, we recommend seriously examining network computing solutions for many users. We also recommend implementing network computing solutions, even for PC users where they make sense, such as Web-delivered Java applications, to help promote reliability within your network.

9.3 A "Fault Tolerant" DHCP Server

DHCP servers must be reliable if the network is to be consistently available. In other words, they simply cannot crash. If they do, you may have a network which starts to fall apart, preventing clients from attaching and getting addresses. Unfortunately, there's nothing in the DHCP RFC which provides for a backup or standby server, so most DHCP servers on the market do not handle server failures, even though such failures can be catastrophic.

While it's certainly possible to purchase true fault tolerant solutions, such as Vinca's StandbyServer for OS/2 Warp (see: <http://www.vinca.com/products/sbsos2/os2data.html> on the Internet) or an IBM AIX cluster (see 9.4, "AIX Features" on page 340), we thought it would be helpful to show a simple standby server solution using the features built into OS/2 Warp Server and TCP/IP Version 4.1 for OS/2. By understanding the basics, you can quickly graduate to more sophisticated solutions if needed.

9.3.1 Prerequisites

To implement a "fault tolerant" DHCP network you should have:

- Two systems running OS/2 Warp Server (with TCP/IP 4.1 and its DHCP server software installed). One of these systems, the backup, may be an older, slower 486, as long as it has just enough performance to keep the network up and running on an emergency basis should the primary server fail.
- The backup DHCP server can, itself, be a DHCP client if it's on the same subnet as the primary. Otherwise, we assume your backup DHCP server has a static IP address assigned on another subnet.
- With multiple subnets, your router(s) should include active BOOTP relay agent(s), so that DHCP requests from clients will be handled by either the primary or backup DHCP server as needed.
- For testing purposes, you may wish to have one DHCP client, of any type, available on the network.

We also assume that your primary DHCP server is up and running correctly and that Boot Manager is installed.

Although optional, we recommend also installing OS/2 Warp Server's NetFinity services (available through IBM Software Choice), at least on your backup server, to process alerts so that you can be notified if the primary DHCP server fails.

9.3.2 Step-by-Step Procedure

1. Enable REXECD and FTPD on the primary server. If REXECD is running on the primary server, tasks can be started on the server from a remote system. Likewise, if FTPD is running on the primary server, then files can be transferred to/from the server across the network using FTP. Naturally, you should password protect both of these services.

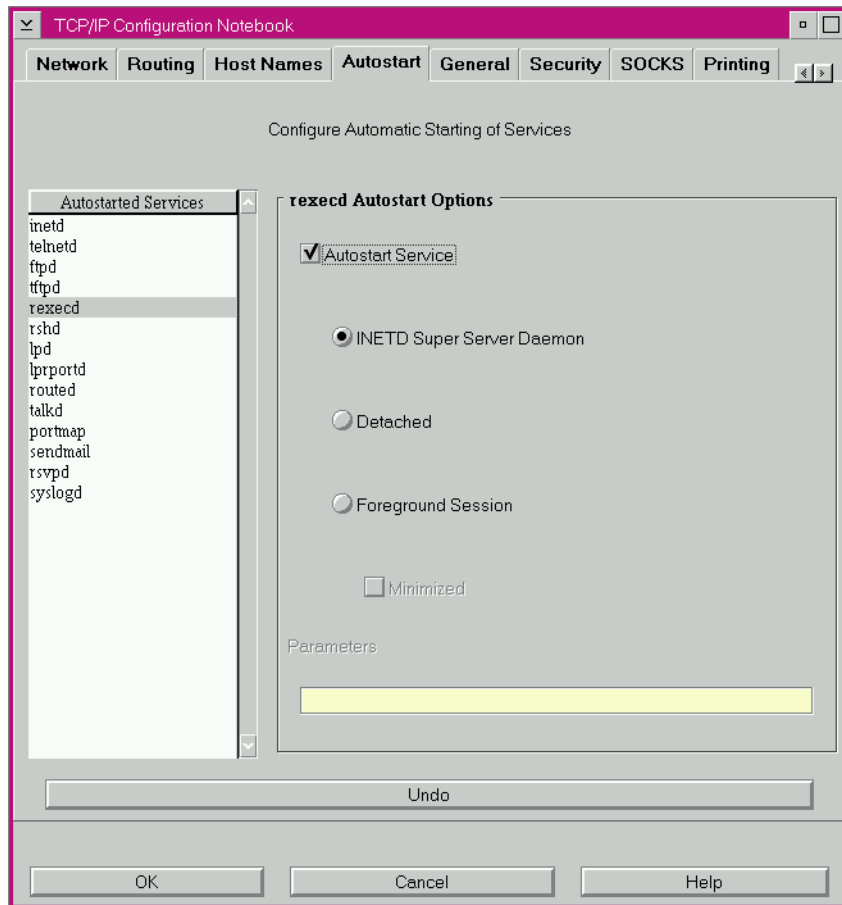


Figure 235. [Warp Server] Choosing Autostart Service for REXECD

The TCP/IP Configuration notebook controls whether REXECD and FTPD will be automatically started and what user names and passwords they will accept. Since the TCP/IP Configuration notebook is a Java application, you can access the notebook at the server console or from any Java-enabled system on the network. As shown in Figure 235, start by clicking on the **Autostart** tab in the TCP/IP Configuration notebook. Then, highlight **REXECD** and make sure the **Autostart Service** checkbox is selected. Repeat the same procedure for FTPD. (Both can be started from INETD; so you shouldn't have to click on any of the other radio buttons.)

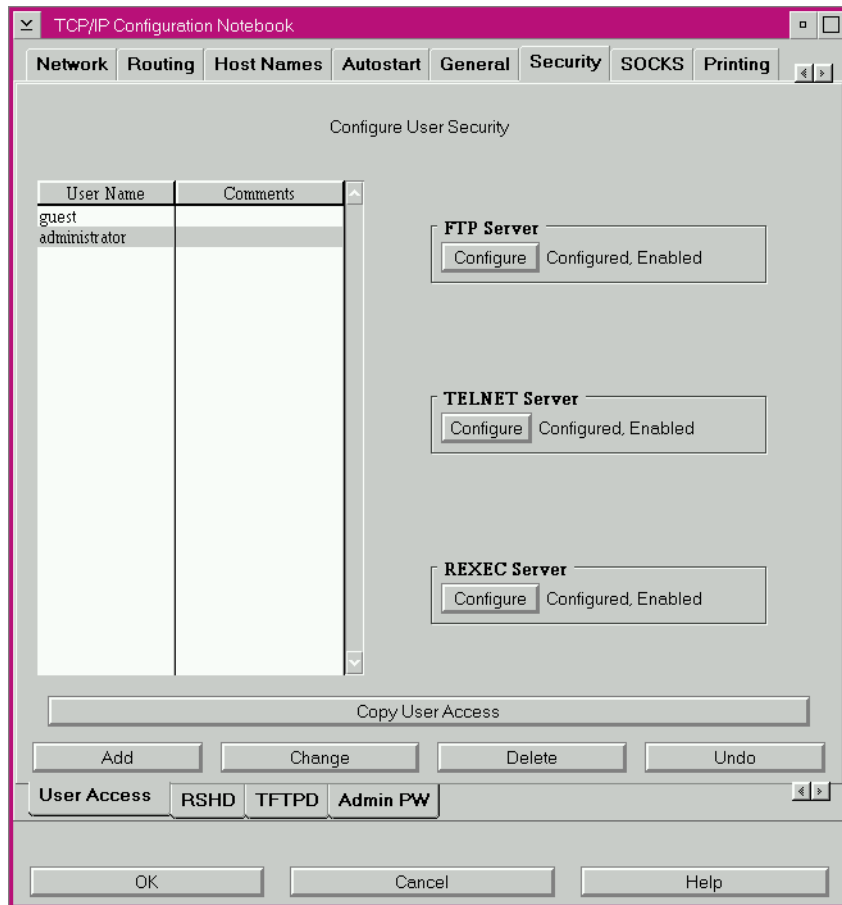


Figure 236. [Warp Server] Security Section in TCP/IP Configuration

Security settings for both FTPD and REXECD can be controlled by first clicking on the **Security** tab (see Figure 236). Add or change users using the list box on the left. In this example, we've defined a user name of administrator with a password of password. Then, click on the buttons for **FTP Server** and **REXEC Server** to enable access to these services for the user administrator. (See Figure 237.) At a minimum, you should provide read-only access to the directory specified by the ETC environment variable on the primary server, normally C:\MPTN\ETC.

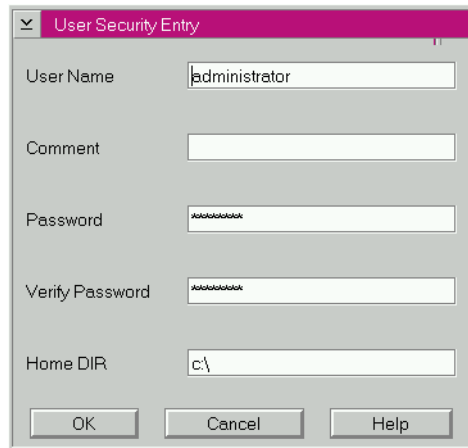


Figure 237. [Warp Server] Configuring REXECD Options

When you've finished making these changes, close the TCP/IP Configuration notebook, save your changes, and restart the server.

When the server restarts, you may wish to verify that REXECD and FTPD are operating properly by accessing the server using FTP and REXEC commands at a client.

2. Copy FAILSAFE.CMD to the backup server and add to its Startup folder. FAILSAFE.CMD runs on the backup server and continuously monitors the primary. You should add a program object to the backup server's Startup folder to automatically run FAILSAFE.CMD each time the backup server boots.

Figure 238 and Figure 239 on page 337 display the REXX program FAILSAFE.CMD.

```

/* FAILSAFE.CMD
 * by Timothy Sipples and Peter Degotardi
 *
 */

/* FAILSAFE is designed to monitor a DHCP server and make sure that
 * it is in continuous operation. If the DHCP server fails then the
 * monitoring system takes over server responsibilities, shuts down
 * the failing server, and sends an alert.
 */

/* The following three values are assumed valid for all needs: */

server_IP = "192.168.7.10"
user_ID = "administrator"
password = "password"
same_subnet = 1
check_every = 30
config_file = "C:\MPTN\ETC\DHCPD.CFG"

address cmd

call RxFuncAdd "SysSleep","RexxUtil","SysSleep"
call RxFuncAdd "FtpLoadFuncs","rxFtp","FtpLoadFuncs"
call FtpLoadFuncs

say "Standing by to respond to any failure of server "server_IP"..."

do forever
  rc = FtpSetUser(server_IP,user_ID,password)
  rc = FtpGet(config_file,config_file,"Binary")
  rc = FtpLogoff()

  "@dadmin -h"server_IP" -u"password" -n > nul"
  if rc \=0 then do
    "@dadmin -h"server_IP" -u"password" -n > nul"
    if rc \= 0 then leave
  end

  say date()" "time()": Check OK!"
  call SysSleep check_every
end

```

Figure 238. [OS/2 Warp] FAILSAFE.CMD (Part 1 of 2)

```

/* Server Failure Detected */

say date() "time()": Houston, we have a problem..."
'@start /n rexec 'server_IP' -l'user_ID' -p'password' SETBOOT /T:NO
/B'
call SysSleep 5

if same_subnet then do
    "@dhcpmon -t"
    "@arp -f"
    "@ifconfig lan0 "server_IP
end

'@del c:\mptn\etc\dhcps.ar'
'@del c:\mptn\etc\dhcps.cr'
'@start "DHCP Server" /n /min DHCPD.EXE'
'@start "DHCP Server Administration" /min DHCPSCPS.CMD'

```

Figure 239. [OS/2 Warp] FAILSAFE.CMD (Part 2 of 2)

At the top of FAILSAFE.CMD, you can change critical values to match those required on your particular network. These values include the numeric IP address of the primary DHCP server, the user ID and password for REXEC and FTP access to the server, whether the backup is on the same subnet (1) or not (0), how often to interrogate the primary server to make sure it's still running, and the full path and file name for the DHCP configuration file.

FAILSAFE.CMD spends most of its time in a do..forever loop. First, the latest DHCP configuration file is copied to the backup server using REXX FTP functions. Then, the backup attempts to run a simple DHCP administrator report on the primary server. If the primary server doesn't answer, a second attempt is made. If both attempts fail, FAILSAFE.CMD assumes the primary server has failed. If at least one attempt succeeds, FAILSAFE.CMD goes to sleep for a number of seconds (the check_every value), and the loop then repeats.

When a primary server failure occurs, FAILSAFE.CMD will perform several tasks. First, an attempt will be made (using REXEC) to disable the primary DHCP server by rebooting it with the SETBOOT command and placing it indefinitely at the Boot Manager menu ("/T:NO"). Then, FAILSAFE.CMD waits 5 seconds, just to give the primary server time to go offline. If the backup server is on the same subnet as the primary, then the backup's DHCP address is relinquished (DHCPMON -t), the ARP table is

flushed, and the backup server assumes the primary server's static IP address.

Finally, FAILSAFE.CMD deletes any old DHCP database files (DHCP.S.AR and DHCP.S.CR) then starts both DHCP.SD.EXE (the DHCP server) and DHCP.S.CPS.CMD (the DHCP administration access task).

At some point in the primary server failure handling in this REXX script, you should add one or more lines to generate an alert to make sure you are notified by pager, e-mail, or autodialing modem that your primary DHCP server has failed. NetFinity's Alert Manager can help with this task.

Note that FAILSAFE.CMD generates on screen status messages periodically. You can log these status messages by redirecting the screen output of FAILSAFE.CMD to a file. For example, you could type in the command:

```
FAILSAFE > DHCPBACK.LOG
```

to capture these messages in the file DHCPBACK.LOG.

Although this approach employs relatively simple TCP/IP technology, it should help provide some added reliability to your network.

9.3.3 Technical Caveats

- We have not included any mechanism for monitoring the backup DHCP server to make sure it does not fail. Failure of the backup could leave your network vulnerable, and you should be alerted should such a failure occur. You may wish to use NetFinity services on the backup server as a monitoring tool.
- We discovered that the backup server will not always start correctly if it uses the full database from the primary DHCP server. Therefore, the full database is not copied by FAILSAFE.CMD.
- Since we are not copying the full DHCP database from the primary server to the backup, new clients attaching to the network when the backup has taken over may receive an IP address assignment already in use by another client. The client may have to reject several offers from the backup DHCP server until an available address is found. This process can delay final attachment to the network for new clients while the backup server is in charge.
- FAILSAFE.CMD does not include some necessary error checking. For example, you may wish to add error checking to the FTP file transfer. We've purposely kept FAILSAFE.CMD short in order to more clearly demonstrate the basic concepts.

- A router or network connection failure, even a temporary one, could cause the backup DHCP server to activate. Therefore, it's possible that both the primary and backup DHCP servers could be simultaneously active. (If the primary DHCP server suddenly becomes unreachable, the SETBOOT command will not be received by the primary, and it won't shut down.) That shouldn't cause any immediate problem, particularly if the network connection between the two subnets remains broken. Again, the goal is simply to preserve reasonable network access until the primary failure can be remedied at a more convenient time.
- We have not included a standby DDNS server example, because we recommend that you simply provide each client (via DHCP) the IP addresses of at least two name servers. Then, clients will automatically use whichever name server is active. If for some reason you do need to monitor your DDNS server(s), then you can use techniques similar to those presented in this section, substituting a NSLOOKUP command for DADMIN.
- We recommend that you separate the primary and backup DHCP servers by some distance, perhaps placing them in separate offices. A fire, flood, or other mishap may only affect one of the two servers if they are not both in the same place.
- Your DHCP servers, particularly your backup server, should be secured in a locked area. FAILSAFE.CMD contains the user name and password for accessing your primary server in plain text. If an unauthorized person gains access to FAILSAFE.CMD, you could expose your primary DHCP server to attack. If you cannot secure your backup DHCP server, you should find some other way of providing FAILSAFE.CMD with the user name and password information it needs to access the primary server.
- You may wish to relocate the FTP transfer of DHCPD.CFG to a REXX script running on the primary server and push the file to the backup server periodically instead of pulling it. That way the primary server can transmit a new copy of the file only when it changes, cutting down on network traffic. The backup server would then be running FTPD instead, further reducing the load on the primary server. Just make sure you manually copy the latest DHCPD.CFG to the backup server before you start running FAILSAFE.CMD for the first time.

You could also use periodic FTP pushes to verify whether the backup server has failed or not, then generate an alert at the primary server if the file transfer does not succeed.

- One backup DHCP server could conceivably monitor more than one primary DHCP server with some careful additions to FAILSAFE.CMD.

- Resetting the IP address of the backup server to match the address of the primary server can be dangerous if the primary server has not actually failed. If your network hardware supports it, you may wish to have `FAILSAFE.CMD` literally disable the port used by the primary server in the event a failure is detected, to make it impossible for the primary server to rejoin the network.

9.4 AIX Features

You may wish to opt for AIX-based servers as your need for a large, reliable, and stable network continues to grow.

IBM's AIX can be made fault-tolerant by including clustering technology, such as High Availability Cluster Multi-Processing (HACMP). One suggestion is to have a pair of AIX machines running HACMP using a set of shared disks that the DHCP and DNS servers write their data to. The HACMP failover scripts on the secondary machine then need to be modified so that the DHCP and DNS servers are started in the event of a failure on the primary machine.

9.5 Shadow IPserver Features

Shadow IPserver can be expanded into a cluster of peer servers to implement redundant DHCP, dynamic DNS, and NBNS services. The IPserver peers are independent servers that may be installed alongside each other or in different locations. Each server coordinates its activities with the other across the network.

The IPserver cluster architecture allows real-time data replication. Fully redundant data distribution to multiple IPservers provides high reliability, tolerance of hardware failure, and the ability to balance the client load across multiple servers.

Chapter 10. Performance

If you're looking to boost the performance and capacity of your TCP/IP network, this chapter should help. This chapter examines how to set lease times and minimize broadcast traffic, among other issues.

10.1 Leases

The lease time implemented in your network will vary depending on how your network is configured, and by the nature of the clients that are attached to it. Here, we examine how you could determine what the lease time should be on your network.

10.1.1 What is a Lease?

The lease time is the time that a DHCP client has to use the parameters supplied to it by a DHCP server. The time itself is one of the parameters passed. At the end of Chapter 2 are packet decodes which show the T1 and T2 parameters.

10.1.2 How Leases Work

When a client receives TCP/IP configuration parameters from a server, it receives as part of those parameters a lease time that defines how long a client is able to use the parameters. See Appendix E.3.2, "Option 51: IP Address Lease Time" on page 433.

On receipt and acceptance of the parameters, two timers, T1 and T2, start to count down. T1 will expire before T2, and T2 will expire before the end of the lease time.

When T1 expires (at 0.5 times the total lease time), the client will try to extend its lease for the current configuration. If the DHCP server has not responded by the time T2 has expired (0.875 times the lease time), the client will then start broadcasting to try and obtain a new lease.

A client is able to terminate a lease without having to wait for the lease to expire, for example when it is shutting down. This frees up the IP address at the DHCP server and makes it available for other clients

10.1.3 Choosing a Lease Time

There are no hard and fast rules for setting lease times. They can vary from site-to-site according to usage patterns, goals, service levels for the DHCP servers, and WAN stability.

In determining the your optimum lease time, ask yourself these questions:

- Do I have more users than IP addresses?

For example, you may have many mobile users coming and going from your location. Each user that leaves your site can still have a valid lease.

- How many clients do I have?

If you have a fairly short lease time and a large number of clients, you may begin to impact the performance of your DHCP server.

- How long will it take to repair or restart a failed DHCP server?

If a DHCP server fails, clients trying to obtain a lease will encounter problems. Clients already active will have problems when their T2 timer expires.

- Will I need to change IP parameters at the client?

Are there parameters that are likely to change that are needed by the clients to work correctly, such as a new default router.

- How reliable are WAN connections?

The affects of a WAN outage on a client are the same as if the DHCP server itself has failed. The server may experience heavy loads when a failed WAN connection is reestablished and all the remote clients try to obtain leases.

- Will I need to reclaim addresses?

Will you need to 'get back' IP addresses so they can be statically assigned to new hosts, such as a new router.

Rules-of-Thumb

Make the lease time twice as long as any potential outage of the DHCP server.

Remember: The longer the lease, the longer it takes for changes in DHCP options to reach the clients.

10.1.3.1 Example Lease Times

Here's a list of example lease times, with the possible reasons you could use them:

15 minutes	Allows you to maximize available addresses when you have a dynamic environment with lots of mobile users and therefore a shortage of IP addresses.
6 hours	Allows you time to repair a failed DHCP server. (3 hours)
12 hours	Allows you to distribute new IP parameters and reclaim an IP addresses overnight.
24 hours	Allows clients to continue working in the morning if a DHCP server has failed overnight.
3 days	Appears to be Microsoft's default value, so is used by many locations.
6 days	Allows clients to continue working on Monday morning if a DHCP server has failed over a weekend.
1 year	If a client has not used the address for a year, they have probably left, and the address can be reused.

10.1.4 Multiple Leases

With two or more DHCP servers on a network, clients that are moved around (for example, laptops) can end up with multiple and redundant leases. Consider a local network with two DHCP servers and a remote site also served by those servers. A mobile client initially connects to the local network and receives a lease from one of the two servers. It is then moved to the remote network without releasing the lease. When it attempts to use the address it already has, it is NAKed by the server, and the client will then receive an address appropriate for the remote network.

If the client is then moved back to the local network and tries to use the address allocated for the remote network it will again be NAKed by the server. Now the client, rather than reusing the lease it originally had for the local network, will broadcast a DHCPDISCOVER to obtain an address. The server that holds the previous lease will offer the address back to the client, but there is no guarantee that the client will accept the address. Therefore, it is possible that the client will obtain an address from the other server and now hold two leases within the local network.

This problem can be eased by using only one DHCP server per network or location and also by using short lease times.

If in your network, DHCP servers are maintained by individual departments, offices, or individuals with their own small address pools, they can find that their addresses are being used by any device on the LAN that has been configured to use DHCP.

10.2 Monitoring and Troubleshooting

Most TCP/IP implementations come with a few standard tools that can be used in monitoring and debugging your IP network. Alternatively, you could use third-party applications and dedicated analyzers to do this more thoroughly.

The following commands are included with most implementations of TCP/IP and can be used to check connectivity. The following commands have been documented using TCP/IP Version 4.1 for OS/2. Their usage may vary with other operating systems.

10.2.1 The PING Command

By sending ICMP echo requests to specific IP addresses, `PING` (short for Packet Internet Groper) is used to verify connections between hosts.

PING Syntax

```
PING [-?drv] host [size [packets]]
```

Where:

- `-?` Displays the syntax of the `PING` command.
- `-d` Bypass the normal routing tables.
- `-v` Verbose output. Include all ICMP packets received.
- host* A host's name, IP address or broadcast address.
- size* The size of data portion of the packet.
- packets* The number of Echo Request packets to send.

For example:

```
[C:\]PING banquet 56 2
PING banquet.AUSTIN.COOKING.NET: 56 data bytes
64 bytes from 192.168.6.1: icmp_seq=0. time=0. ms
64 bytes from 192.168.6.1: icmp_seq=1. time=0. ms
---banquet.AUSTIN.COOKING.NET PING Statistics---
2 packets transmitted, 2 packets received, 0% packet loss
```

round-trip (ms) min/avg/max = 0/0/0

10.2.2 The TRACERTE Command

TRACERTE (or `tracert` on some other platforms) is used to trace the route an IP packet takes to reach a host on a remote subnet. It can tell you where and why a route is lost.

TRACERTE Syntax

```
TRACERTE [-dnrv] [-w wait] [-m max_ttl] [-p port#] [-q nqueries]
[-t tos] [-s src_addr] host [data_size]
```

Where:

<code>-d</code>	Turn debug on.
<code>-n</code>	Display IP addresses instead of host names.
<code>-r</code>	Disables routing of IP packets.
<code>-v</code>	Turns verbose on.
<code>-w wait</code>	Wait time in <i>wait</i> seconds between packets.
<code>-m max_ttl</code>	Maximum time-to-live in <i>max_ttl</i> hops.
<code>-p port#</code>	Destination port number <i>port#</i> .
<code>-q nqueries</code>	<i>nqueries</i> number of probes for each TTL.
<code>-t tos</code>	<i>tos</i> type of service.
<code>-s src_addr</code>	Source IP address <i>src_addr</i> .
<i>host</i>	A host's name or IP address.
<i>data_size</i>	The number of bytes of data used.

For example:

```
[C:\]tracerte 192.168.7.10
tracert to 192.168.7.10 (192.168.7.10), 30 hops max, 38 byte packets
 1 192.168.6.1 (192.168.6.1) 0 ms 0 ms 10 ms
 2 brie (192.168.7.10) 0 ms 0 ms 0 ms
```

TRACERTE returns an indication as to errors encountered while trying to trace the route to the destination system. The indicators are:

!	Port unreachable, connection refused.
!N	Network unreachable, no route to host.
!H	Host unreachable, no route to host.

!P Protocol unreachable, connection refused.
!F Message to big.
!S No route to host.

For example:

```
[C:\]tracerte 192.168.10.1
tracert to 192.168.10.1 (192.168.10.1), 30 hops max, 38 byte packets
 1 192.168.6.1 (192.168.6.1) 10 ms 10 ms 0 ms
 2 192.168.6.1 (192.168.6.1) 0 ms !H 0 ms !H 0 ms !H
```

Indicates that our default router 192.168.6.1 has no routing information to reach the host 192.168.10.1.

10.2.3 The IPTRACE Command

IPTRACE is used to trace all packets received by and sent from a network interface. All data collected will be written to the **IPTRACE.DMP** file in the current directory. Depending on the platform, the dump file may be stored in a non-editable format. In OS/2, the **IPFORMAT** command is used to view the dump file; in AIX use the **IPREPORT** command.

IPTRACE Syntax

```
IPTRACE [-i] [interface]
```

Where:

-i Specifies that only IP packets should be traced.
interface Specifies an interface to be traced.

10.2.4 The ARP Command

ARP is a mechanism to dynamically map IP addresses to the MAC addresses of other network adapters in the same local subnet and then cache them in memory for future reference. You can use the **arp** command to display and manually maintain the ARP cache.

arp Syntax

```
arp [-afds?] hostname [hardware_addr] [temp|pub]
```

hostname A host's name or IP address.

-? Show the online help.

-a	Print all <code>arp</code> table entries.
-f	Flush all <code>arp</code> table entries.
-d	Delete <code>arp</code> table entry for the host <i>hostname</i> .
-s	Add <code>arp</code> table entry for the host <i>hostname</i> .
temp	Timeout this entry if it is not used.
pub	Reply for other host.

10.2.5 The NETSTAT Command

The `NETSTAT` command displays the network status of the local workstation. It supplies information about routing, TCP connections, UDP statistics, IP statistics, memory buffers, and sockets.

NETSTAT Syntax

```
NETSTAT [ -? ] [-acghilmprstu]
```

Where:

- a Displays addresses of network interfaces.
- c Displays ICMP statistics.
- g Displays IGMP statistics.
- h Displays the resolved host name.
- i Displays IP statistics.
- l Displays information about the socket that is listening.
- m Displays information about memory buffer usage.
- n Displays information about LAN interfaces.
- p Displays the contents of the ARP table.
- r Displays the routing tables.
- s Displays information about sockets.
- t Displays information about TCP connections.
- u Displays UDP statistics.

For example, to dump the current routing table:

```
[C:\]NETSTAT -r
destination          router                netmask              metric flags intrf
```

default	192.168.6.1	0.0.0.0	0	UGP	lan0
127.0.0.1	127.0.0.1	255.255.255.255	0	UH	lo
192.168.6	192.168.6.10	255.255.255.0	0	UC	lan0
192.168.7.10	192.168.6.1	255.255.255.255	0	UGHW3	lan0
192.168.9.10	192.168.6.4	255.255.255.255	0	UGHDM	lan0

10.2.6 The HOST Command

The `HOST` command is used to ask a DNS server to resolve hostnames to IP addresses and vice versa.

HOST Syntax

```
HOST <hostname>
```

hostname A host's name or IP address.

10.2.7 The NSLOOKUP Command

`NSLOOKUP` is used to query DNS servers in either an interactive or noninteractive fashion.

NSLOOKUP Syntax

```
NSLOOKUP [-options]           (Interactive mode using default server)
[-options] -server             (Interactive mode using server)
[-options] host                (Look up host using default server)
[-options] host server         (Look up host using server)
```

where *[options]* are:

<code>all</code>	Print options, current server and host.
<code>[no]debug</code>	Print debugging information.
<code>[no]d2</code>	Print exhaustive debugging information.
<code>[no]defname</code>	Append domain name to each query.
<code>[no]recurse</code>	Ask for recursive answer to query.
<code>[no]search</code>	Use the search list.
<code>[no]vc</code>	Always use a virtual circuit.
<code>domain=name</code>	Set default domain name to <i>name</i> .
<code>port=X</code>	Use TCP/IP port number <i>x</i> .
<code>srchlist=n1[/n2/.../n6]</code>	Set domain to <i>n1</i> and search list to <i>n1</i> , <i>n2</i> , etc.

<code>root=<i>name</i></code>	Set root server to <i>name</i> .
<code>retry=<i>X</i></code>	Set number of retries to <i>x</i> .
<code>timeout=<i>X</i></code>	Set initial time-out interval to <i>x</i> seconds.
<code>querytype=<i>X</i> or type=<i>x</i></code>	Set query type, for example: A, ANY, CNAME, NS, PTR..
<code>class=<i>x</i></code>	Set query class to one of IN (Internet), CHAOS, HESIOD or ANY.

10.2.8 Other Utilities

Here (briefly) are some tools that you could consider using to monitor and troubleshoot your network.

10.2.8.1 NetFinity

NetFinity Manager and Client Services for Netfinity Manager is not a tool for administering TCP/IP. Rather, it is a systems management tool that allows you to monitor, manage and configure both local and remote systems.

Netfinity can be used over a variety of transport protocols such as TCP/IP, SNA and NetBIOS.

Part of Netfinity - the Systems Monitor Service - allows you to collect statistics from active systems. For example CPU, memory, disk and print usage can be monitored. You can also use the Systems Monitor Service to collect TCP/IP statistics.

See Part C.3, "NetFinity" on page 406 for information regarding running Netfinity in a DHCP environment.

10.2.8.2 Third-Party Alternatives

There are many third-party network monitoring and analyzing tools available. These can range from software that can be run on existing PCs on the network, through to dedicated devices that attach directly to the network.

Current examples include:

- Fluke Corporation supplies several diagnostic solutions such as OneTouch, Enterprise LANMeter and Network Inspector.
- Hewlett Packard has its NetMetrix range.
- Network Associates has available the software-based Sniffer Basic and the dedicated Sniffer Pro LAN.

10.3 Troubleshooting TCP/IP Networks

Sooner or later, you will encounter problems on your network. How quickly you can resolve these problems depends on your approach to troubleshooting. Here we examine ways to debug your network using the commands that come as standard with most implementations of TCP/IP.

10.3.1 Prerequisites for Troubleshooting

If you encounter connectivity problems with your network, it may take many steps to find the problem, and hence the solution. When debugging a network problem, keep the following prerequisites in mind.

Understand TCP/IP

To solve an IP related problem, it goes without saying that you have to have a high understanding of TCP/IP. TCP/IP is an open, multi-vendor protocol. As such, there is a lot of publicly available information available – see your local bookstore.

Know Your Environment

As a network administrator, you should know what equipment and systems have been placed onto the network. Not only should you know what equipment is out there, you should know and understand their role within the greater whole.

Any Information Can Help

Any symptom can be a clue to currently occurring problem, even if at first glance it appears to be unrelated.

Don't Believe What People Say

Don't believe what people have told you until it has been verified.

You Can be Limited by the Person On-Site

Your current skill level is equal to that of the person on the other end of the phone. Unless you can talk them through your diagnostic procedures, diagnosis may not start until you can actually get on-site.

10.3.2 A Bottom-Up Approach

Because protocols are divided into several layers, and each layer's connectivity depends on the layer beneath it, it therefore is reasonable to start diagnosing network problems from the bottom and work our way up.

TCP/IP does not match the OSI seven-layer model completely, but it is useful to use the following definitions. Each layer is explained from the bottom to the top.

10.3.2.1 Layer 1 Physical Layer

The physical layer represents the media used within the network, such as fiber optic or coaxial cable. It is only responsible for the transmission of data across the physical network.

10.3.2.2 Layer 2 Data Link Layer

The data link layer represents a communication link between two systems. Network types such as Ethernet, token-ring and FDDI are covered by this layer. The data processed in this layer is often called a frame.

In debugging the network layer, you will probably use the `arp` command as documented on page 346. Keep the following in mind:

- When `arp` fails, you cannot communicate.
- When `arp` succeeds, you don't have any hardware problems.
- When `arp` succeeds, but you still can't communicate, then you have a problem in a higher layer.

The `arp` command cannot be used to diagnose higher layers, but applications that use the higher layers (such as `PING`) can assist in diagnosing lower layers.

In using `arp` for testing, use the following procedure:

1. Clear the ARP cache with the an `arp -f` command.
2. Check that the cache is empty with an `arp -a` command.
3. Try to `PING` the host you are trying to connect to.
4. Then check the ARP cache again. If the `PING` was successful, and the host you `PINGed` is on the same subnet, there will be an entry for the host in the table. If the host is on another subnet, there should be an entry for the router that would be used. (Note in the following example that an ARP entry for our nameserver with the address 192.168.6.5 was also added.)

```
[C:\>]arp -a
        ARP table contents:
interface      hardware address      IP address      minutes since
                        last use
lan0           0 :6 :29:b3:e :ed        192.168.6.1      0
lan0           8 :0 :5a:ce:ea:cb        192.168.6.5      0
```

If the `PING` was unsuccessful, and the host is on the same subnet, try to `PING` other devices on the subnet. If the host being `PINGed` is on a different subnet, try `PINGing` the default router. By doing this, you can receive some indication as to where to next proceed.

Use `arp` when checking for duplicate IP addresses. If the ARP entry for an IP address is different when queried from different locations, then you know that you have a duplicate address.

10.3.2.3 Layer 3 Network Layer

The network layer represents communication between multiple systems. In TCP/IP, IP and ICMP are network layer protocols, but IP is the only protocol to carry user data. The data passed through this layer is often called a packet. (An IP packet is sometimes called an IP datagram). The IP address is used to identify various systems within the network.

This layer also allows for the connection between two systems, and between two systems by passing through other systems, through a process known as routing.

In debugging the network layer, there are several commands that can be useful.

Use the PING Command

The `PING` command is documented on page 344. When using `PING`, keep the following in mind:

- When `PING` fails, you cannot communicate by using the IP protocol
- When `PING` succeeds you don't have hardware or network configuration problems.
- When `PING` succeeds, but communication still fails, there is an application problem.

If you cannot connect to a host, try the following `PING` sequence to help determine where the problem lies:

1. `PING` the IP address 127.0.0.1 by issuing the command

```
ping 127.0.0.1
```

As discussed on page 5, 127.0.0.1 is the loopback IP address. If you can successfully `PING` this address, then you have proven that the IP stack on your host is working correctly.

2. `PING` your own IP address. A `PING` to your own address will physically transmit out onto the network. If the `PING` is successful, then you know that your network adapter is working.
3. `PING` the IP address of your default router. This will determine if your default router is connected to the network, but not that it is working correctly. It can also show if your subnet mask is correct.
4. `PING` an IP address of a host on the same subnet of the host you want to connect to. This will determine if all routers between you and the host you want to connect to are working correctly.
5. `PING` the IP address of the host you want to connect to.

In the above procedure, we have used IP addresses only because any problems in communicating with your name server, or misconfigurations in your name server, can cause connectivity problems in general when you are trying to connect by name.

In the following example, we are trying to `PING` the machine called hotdog:

```
[C:\]PING hotdog 56 2
PING hotdog.AUSTIN.COOKING.NET: 56 data bytes

----hotdog.AUSTIN.COOKING.NET PING Statistics----
2 packets transmitted, 0 packets received, 100% packet loss
```

As you can see, the `PING` command itself has failed, but the host name was successfully resolved by the name server. This proves that the name server is working, although the address supplied by the name server may not necessarily be correct.

Use the TRACERTE Command

If in step 4 above, if you could not `PING` any hosts on the remote subnet, you can use the `TRACERTE` command as documented on page 345 to determine where and why data is being lost.

Use the NETSTAT Command

Use the `NETSTAT` command as documented on page 347. Use `NETSTAT -i` to check IP statistics, and `NETSTAT -c` to check ICMP statistics. Figure 240 on page 354 shows output from the `NETSTAT -i` command.

```
[C:\]NETSTAT -i
total packets received 5548
checksum bad 0
packet too short 0
not enough data 0
ip header length < data size 0
ip length < ip header length 0
fragments received 0
frags dropped (dups, out of space) 0
fragments timed out 0
packets forwarded 0
packets rcvd for unreachable dest 0
packets forwarded on same net 0
Unknown/Unsupported protocol 2
requests for transmission 4974
lost packets due to no bufs, etc 0
output packets discarded because no route could be found 0
input packets delivered successfully to user-protocols 5546
input packets with an unknown protocol 0
output packets successfully fragmented 0
output fragments created 0
fragmentation failed 0
successfully assembled packets 0
Packets received with version !=4 0
Raw ip packets generated 193
```

Figure 240. Output from NETSTAT -i

Look for unusual counters. If output packets discarded because no route could be found is not zero, you have a routing problem somewhere. If NETSTAT -c gives a high Source Quench count, the host you are talking to (or the routers you are talking through) may be overloaded.

10.3.2.4 Layer 4 Transport Layer

This layer represents a connection between two processes. Any system can have multiple processes running on it, and TCO and UDP are the protocols used to achieve this connection. TCP data passed through this layer is often called a segment; UDP data is known as a datagram.

This layer also provides functionality for flow control and reliability (including retransmission). TCP provides these functions, while UDP does not.

Use the NETSTAT Command

Use the NETSTAT command as documented in 10.2.5 on page 347. Use NETSTAT -t to check TCP statistics and NETSTAT -u to display UDP statistics. Additionally, use NETSTAT -s to check socket information. Figure 241 on page 355 shows part of the output from the NETSTAT -t command.


```
[C:\>]NETSTAT -t
TCP STATISTICS
connections initiated          2
connections accepted          1
connections established        2
embryonic connections dropped  0
conn. closed (includes drops)  4
segs where we tried to get rtt 10
times we succeeded              8
delayed acks sent              8
conn. dropped in rxmt timeout  0
retransmit timeouts            0
persist timeouts               0
keepalive timeouts             0
keepalive probes sent          0
connections dropped in keepalive 0
total packets sent             23
data packets sent              6
data bytes sent                92
data packets retransmitted     0
data bytes retransmitted       0
ack-only packets sent          12
window probes sent             0
packets sent with URG only     0
window update-only packets sent 1
```

Figure 241. Output from NETSTAT -t

Again, look for unusual counters. If `packets received with cksum errs` is not zero, you are experiencing errors somewhere on your network.

10.3.2.5 Layer 5 Session Layer

The session layer provides dialog sessions such as full- and half-duplex and synchronization points in the dialog. Within TCP/IP, there is no precise session layer, although some of its functionality is provided within TCP.

10.3.2.6 Layer 6 Presentation

The presentation layer provides for common data presentation between applications. Within TCP/IP, there is no precise presentation layer.

10.3.2.7 Layer 7 Application.

The application layer represents the application entity, usually an application program.

When diagnosing DNS issues, there are two commands that are useful.

Use the HOST Command

The `HOST` command, as documented on page 348, is used to query a DNS server and can be used to verify that your DNS configuration is correct. `HOST` can only be used for name to address and address to name resolution.

Use the NSLOOKUP Command

The `NSLOOKUP` command, as documented on page 348, is used to query a DNS server and can be used to verify that your DNS configuration is correct.

The reader should also read the *TCP/IP Command Reference* in the TCP/IP Information folder of Warp Server.

This section provides an overview of how to use `NSLOOKUP`. The information presented here should work for both OS/2 and many UNIX versions of `NSLOOKUP`. All Windows-based versions of `nslookup` that we have seen are GUI programs and have different (and varying) usage.

`NSLOOKUP` is a resolver that sends queries to name servers. This is the same task that your browser has to do when locating a particular Web page (if you entered the server name rather than the server IP address). By default, `NSLOOKUP` performs recursive queries. See Figure 242 on page 357 for an example of how a name server resolves a recursive query. Resolvers make recursive queries which causes the name server the most amount of work. Name servers themselves make nonrecursive queries. You can configure `NSLOOKUP` to do nonrecursive queries as well.

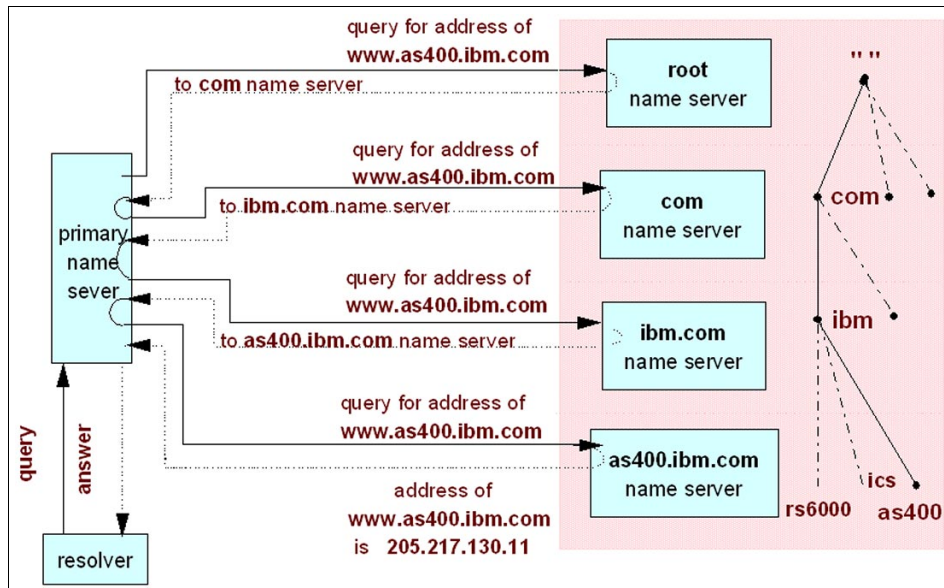


Figure 242. Name Resolution Example

`NSLOOKUP` is a useful while you are configuring your name server(s) and also for troubleshooting certain issues after configuration. For example, say you have configured one of your servers to provide zone transfers to only a specified group of machines (for example, by using the `secure_zones` option). It is probably prudent to use `nslookup` to try a zone transfer (using the `ls` subcommand) from a machine other than the trusted machines, rather than to wait for crackers to test your configuration for you. Most name servers, you will find, will not pass you zone information for security reasons.

`NSLOOKUP` can be run interactively, or if you have a single query, you can type the query from the command line. When you start `NSLOOKUP` interactively, it is useful to enter the command `set all` to remind you of your default settings (particularly your default name server, domain, and search list, if any). To finish an interactive session, type `exit` or press **CTRL+Z** (end of file on OS/2) or **CTRL+D** (end of file on UNIX). For extra help during an interactive session, type `?` or `help` at the `>` prompt.

`NSLOOKUP` can be configured for recursive and nonrecursive queries by using the `set recur` and `set norecur` subcommands. You can start the program so that it uses the default name server, or you can direct it to use a specific name server for your session. The file `%ETC%\RESOLV2` will contain the name of your default name server (if any), and your default domain. An

example of %ETC%\RESOLV2 is shown below for a machine on our armonk.cooking.net subnet.

```
domain armonk.cooking.net
nameserver 192.168.6.10
```

You could include the line `options debug`, which turns on debugging for `NSLOOKUP`. When you first start using `NSLOOKUP`, it may be useful to have debug on to better understand what is happening. With `debug on` (or `d2`), you will see just about as much DNS information as you would see if you were using a protocol analyzer. We are not using `debug` here because of the amount of information it creates. We do suggest that you use `debug`, though, while you are getting familiar with `NSLOOKUP`.

The `NSLOOKUP` syntax on page 348 shows how to specify a particular name server to use. For example

```
NSLOOKUP - 192.168.7.10
Server: fajita.austin.cooking.net
Address: 192.168.7.10
> exit
```

This starts `NSLOOKUP` using the FAJITA name server, rather than our default name server merlot.armonk.cooking.net. This is useful when your default server is not available due to network or other problems. You will know there is a problem with your default name server if `NSLOOKUP` exits immediately with an error message indicating:

- Timed out
- No response from server
- Connection refused
- Server failure
- Default servers not available

If you do get errors like these, you probably want to start `NSLOOKUP` using a different name server, until you can determine what is wrong that the default name server.

Do not confuse the previous errors with errors of the type:

- Nonexistent domain (host or domain is not known to name server)
- Query refused (often because of security settings at name server)
- Format error (possible error in `NSLOOKUP` or network packet error; the name server thinks request packet is an improper format)

- No records (of the type you are requesting; change your `querytype` to `any` and try again)

You can also redirect some `NSLOOKUP` output to a file. For example, when in interactive mode if you wanted to create a file of all mail exchanger (`MX`) records (that is, mail hosts) for the domain `armonk.cooking.net`, you would do the following commands (the `t` is for recordtype):

```
nslookup
Server: fajita.armonk.cooking.net
Address: 192.168.6.10
> ls -t MX armonk.cooking.net > mailhost.txt
> view mailhost.txt
```

While records are still being read, you will see a series of `#` signs going across the screen; so you know something is happening. The `view` command shows you the file as if you were using the `more` command (which is what `view` uses). If you want to halt a long command, you can type `CTRL+C` during the command, which will bring you back to the `>` prompt. For example, if you entered the `ls -t MX` command on a large Internet domain, it would possibly generate pages of information.

```
> ls -t any armonk.cooking.net
```

This would show all records for the domain, which would, of course, give even more information than just the `MX` records. The record types that you can show individually are:

<code>A</code>	Hostname to IP address records.
<code>ANY</code>	All records available from name server.
<code>CNAME</code>	Canonical names for aliases. For example, we use <code>ns-updates</code> as an alias for the canonical name of <code>fajita.armonk.cooking.net</code> . Usually you will see <code>mail</code> or <code>www</code> as an alias for the fully qualified domain name of the server the mail server or Web server runs on.
<code>HINFO</code>	Host information.
<code>KEY</code>	Public key information.
<code>MB</code>	Mailbox information.
<code>MINFO</code>	Mailbox or mail list information.
<code>MX</code>	Mail exchanger.
<code>NS</code>	Name server.
<code>PTR</code>	IP address to hostname records.

SIG	List records with expiration time and signature information such as A and KEY records.
SOA	Domain's start of authority.
TXT	Text records.
UINFO	User information. For example, will sometimes contain contact information of responsible person for a particular server.
WKS	Well-known services that the host advertises.

The record types are discussed in more detail in 4.4.12, "Record Types" on page 99.

When you enter the `querytype` to `nslookup` subcommands, remember that they are case insensitive. The `ls` subcommand causes zone transfers, and some hosts will not allow that (for security or because of the extra workload on the name server and network).

The `set` command is used when in the interactive mode to change default settings. For example, it can be useful to do a `set debug` or `set d2` (both of which turn on debugging) so that you can see how the name resolution is occurring when you make a query. You can change the default `querytype` for a single command. For example, to show all name server records for `armonk.cooking.net` domain, enter:

```
[C:\]NSLOOKUP -query=NS armonk.cooking.net
Server: fajita.armonk.cooking.net
Address: 192.168.6.10
armonk.cooking.net nameserver = fajita.armonk.cooking.net
armonk.cooking.net nameserver = ns-update
fajita.armonk.cooking.net internet address = 192.168.6.10
ns-update internet address = 192.168.6.10
```

Your default record query type can be set to `NS` as opposed to the default of any records:

```
[C:\]NSLOOKUP -query=NS
Default Server: fajita.armonk.cooking.net
Address: 192.168.6.10
>
```

We are now in the interactive mode and can give more commands. To change our default name server:

```
> set root=192.168.7.10
>
```

To change our default `querytype` back to `any` do

```
> set q=any
```

We can abbreviate most subcommand arguments using the first unique character pattern of the name. Now we would be using the name server for our `austin.cooking.net` subnetwork. This could become confusing because our default domain is still `armonk.cooking.net`, so we might want to use a search list to include both domains.

```
> set srchlist=austin.cooking.net/armonk.cooking.net
>
```

Now, when the name resolution occurs, first `austin.cooking.net` will be appended to the name we are looking up, then `armonk.cooking.net` will be appended, then the name will be searched by itself with no appending. For example:

```
> jalapeno
```

Name resolution occurs as follows. When `defname` and `search` are both on (the default), `NSLOOKUP` uses the search list and successively appends each name in the list to any host query not ending in a dot until the name resolution is successful (`jalapeno.austin.cooking.net` and then `jalapeno.armonk.cooking.net` would be searched). If no attempt is successful, `NSLOOKUP` tries the user-entered query as-is. If you do not want the search list to be used you can append a dot (`.`) to your command as follows:

```
> jalapeno.austin.cooking.net.
```

and then the name is used as is with no appending. Note that the name resolution stops when an address is found. If `search` is off and `defname` is on, `NSLOOKUP` does not use the search list, but appends only the domain name (which is the first entry in the search list). Also, `NSLOOKUP` will only append this domain name to user queries which do not have dots in the name, (rather than names which do not end in a dot, which was the case when `search` was on).

If `defname` is off, no domain is appended to the query. In this case, setting `search/nosearch` has no effect.

10.4 Tuning TCP/IP Networks

Once you have our network running reliably, you will probably want to tune it so that it runs to its greatest potential.

10.4.1 An Approach to Tuning Your Network

There is no set method to performance tuning, and as such, in this section we discuss one approach to tuning.

Gather Information

Gather as much information as possible so that possible causes to your problem (in this case poor performance) can be found. Some questions to ask yourself are:

- Do all applications suffer from poor performance?
- Do all operations within an application suffer from poor performance?
- Do all systems within the network suffer from poor performance?
- When did the problem begin?
- Is the problem intermittent?
- Are there any error messages on either the console or logged to file somewhere?

Understand Your Environment

You have to know and understand the environment that you are dealing with. This includes all products – both hardware and software – and protocols that actively use the network.

Check that all products are being used within specification. For example, most networking environments specify the maximum number of workstations and maximum cable lengths that can be used in a network segment. If additional workstations have been added to the network, this maximum may have been exceeded.

Is it Really a Problem?

Are your users expecting too much? If they are used to loading files from a local file server, they may start to complain that it's too slow when downloading from the Internet! A T1 connection is not going to give the same throughput as a 16 Mb token-ring.

Is it Really a Network Function?

Don't forget that memory usage, disk, CPU, and I/O activity on both the source and target hosts can influence performance. Before you delve deeply into the network, confirm that these resources are not constrained on your hosts.

What do you Want to Achieve?

Are you trying to achieve a better response time, or greater throughput in general? Tuning the response time may entail tuning a particular application on a host, whereas to get better throughput, you might have to visit every workstation, host and router.

10.4.2 TCP/IP Tuning Parameters

When tuning your network, there are several parameters that you may be able to manipulate to enhance network performance. The ability to change these parameters is dependent on the implementation of TCP/IP that you are using.

10.4.2.1 MTU and Fragmentation

The Maximum Transfer Unit is a very important parameter. As TCP/IP works in almost all situations without modifying the default MTU, the MTU is not often tuned. By changing the MTU size, you can minimize the fragmentation that occurs on a packet as it travels through your IP network.

Consider the following:

- When you are sending small packets, and there is no fragmentation:
 - More packets have to be sent; so there is an increased number of I/O operations to the network. This adds overhead.
 - There is lower throughput.
- When you are sending large packets, and there is no fragmentation:
 - Fewer packets have to be sent; so there is a reduced number of I/O operations to the network. This reduces overhead.
 - There is higher throughput.
- When you are sending large packets, and there is fragmentation:
 - More packets have to be sent because the original data has been fragmented. This increases I/O operations to and from the network.
 - There is lower throughput.
 - More packets must be transmitted to recover from lost data packets.

Given the above, we can see that we should set the MTU size to be as large as possible so that a packet travelling through the network does not become fragmented.

You can use the `PING` (see page 344) and `TRACERTE` (see page 345) commands to determine the bottlenecks that occur on your network.

Use `TRACERTE` to determine the IP addresses of the routers between you and the host you are connecting to. Then `PING` each IP address found with a range of packet sizes (use the same range for each address) and graph the results. Any bottlenecks with larger MTU sizes should be readily apparent.

Remember to make the MTU size on your workstation larger than the packet sizes you are using so that fragmentation does not occur before the data reaches the network!

To change the MTU size, you would usually use the `IFCONFIG` command, although some implementations of TCP/IP may use configuration files to hold the MTU definition.

For example, under OS/2 you could issue the commands:

```
IFCONFIG lan0 down
IFCONFIG lan0 mtu 4400
IFCONFIG lan0 up
```

This will set the MTU size to 4400 bytes. Now issue the command:

```
NETSTAT -n
```

This will confirm that the change has been made.

10.4.2.2 TCP Maximum Segment Size

As stated previously, a unit of TCP transmission is known as a segment. IP and UDP datagrams each have a maximum length of 65,536 bytes. A TCP segment also has a maximum length, but it is determined when two systems connect. The Maximum Segment Size or MSS is negotiated by both systems to guarantee connectivity.

The MSS is calculated using the MTU as follows:

1. Calculate variable A.
$$A = \text{MTU} - (\text{TCP header size} + \text{IP header size})$$
2. Calculate variable B.
$$B = \text{Socket Receive Buffer size} / 2$$
3. The smaller of A and B is used as the effective MSS.

Now consider the following scenario. Your workstation is connected to a token-ring network and communicates with hosts also attached to the ring. To enhance performance, you are using a large MTU of 17960 bytes. Later, you have to connect over a router to hosts attached to an Ethernet network. To avoid fragmentation at the router you should set your MTU to 1500 bytes. But

now you have an inefficiency when talking to hosts connected to the token-ring.

How do you resolve this problem? Some stacks have MTU discovery enabled, thereby bypassing the issue. If they don't, you can use the `-mtu` option with the `ROUTE` command. The `-mtu` option allows you to configure the MTU for each route, not just for each interface. It allows you to give a specific value to only the traffic that is passing through a router.

For example, we have workstations on our 192.168.6 token-ring network. They are using an MTU of 4400 bytes. If they need to connect to the 192.168.9 Ethernet network, we could issue the command:

```
route add -net 192.168.9 192.168.6.4 -mtu 1500
```

Here, we are assuming that 192.168.6.4 is the router to the 192.168.9 network. By adding this specific MTU size for traffic to the Ethernet network, we can now communicate efficiently with both local and remote hosts. Our MTU and MSS would be large, however, without fragmentation.

10.4.2.3 The IP Queue

IP keeps a queue for incoming IP datagrams. The queue is only used for receiving datagrams, not sending.

Since routing is done at the IP layer, the IP queue is important in a system configured as a router because it is used when passing datagrams from one network to another. If a router receives a burst of IP datagrams, it is possible for the queue to overflow.

Many implementations of TCP/IP do not allow you to set the queue size. AIX, for example, does.

To set and check the IP queue:

1. To check the current queue size, issue the command:

```
no -o ipqmaxlen
```

2. To set the IP queue length, issue the command:

```
no -o ipqmaxlen=x
```

Where *x* is the new size of the queue.

3. To check for IP queue overflows, issue the command:

```
netstat -p ip
```

The following is an extract from the output, with the counter we are interested in highlighted:

```
...
0 path MTU discovery memory allocation failures
0 ipintrq overflows
0 with illegal source
```

Note that a larger queue may require more CPU time to process; so don't make it too large. If you must increase it, make the increments moderate.

10.4.2.4 Buffers

The buffers are temporary data storage used to store data sent between an application and the network.

Some applications can set their buffer size (by using the `setsockopt()` function), but generally you cannot modify the buffer size.

You can the buffers with the `NETSTAT` command.

`NETSTAT -m` displays details for memory buffer usage.

`NETSTAT -c` displays ICMP statistics. When looking at the ICMP statistics, look for high Source Quench counters.

If an ICMP Source Quench message is received from a router, it means that the router does not have the buffer space needed to queue the datagrams for output to the next network.

10.5 Bandwidth Efficiency

While the easiest solution to network performance problems may be to provide a faster connection, it's not always the most practical. After all, it might not be cost effective to link every remote user or office with a T3 line. Sometimes it pays to closely examine the amount and nature of the network traffic flowing through your network connections to see if it's possible to use limited bandwidth more efficiently.

10.5.1 Broadcast Traffic

Be aware that your network may have multiple protocols running over it. Not all protocols are efficient in how they use the network. Some, like NetBEUI for example, introduce a large quantity of broadcast traffic.

Your environment may use combined bridge/routers (brouters) to interconnect remote locations. If these connections are maintained over relatively slow WAN links, your WAN link may be swamped with broadcast

traffic. In extreme cases, connections that have timer-critical response times (for example SNA), may fail.

As we have seen in Chapter 5, "Integrating File and Print Services" on page 177, it is possible to remove NetBEUI from your WAN connections entirely by implementing TCPBEUI. You can choose to still use NetBEUI within the local network where it is most efficient, but your WAN connections will have been freed up from carrying broadcast traffic.

10.5.2 RSVP

Unfortunately, TCP/IP has not had any standard method of prioritizing traffic until quite recently. Yet, in the real world, bank transactions can be much more important than casual e-mail. Unfortunately, most TCP/IP networks can't tell the difference between high priority datagrams and lower priority ones; so the bandwidth gets split more or less evenly between competing users.

RSVP (Resource ReSerVation Protocol) has been introduced as a way to solve the problem. If your network application has been programmed to take advantage of RSVP, and if the routers enroute support RSVP, then TCP/IP datagrams can be prioritized according to their relative importance. Datagrams for bank transactions can be "tagged" by the application as high priority, while e-mail applications may tag their datagrams as lower priority. Intermediate network devices, such as routers, can then handle the traffic more intelligently, giving more bandwidth to the higher priority traffic as needed.

There's one major downside to RSVP, at least at this point in history: Many applications and network devices do not yet support this protocol extension. Hopefully the widespread adoption over time of IPv6 will help speed the implementation of RSVP as well. Nonetheless, if you have control over your own TCP/IP intranet, and you can take advantage of the RSVP programming interfaces now becoming more and more common, you can design your own network and applications to take advantage of prioritization. Doing so can help you avoid buying more high cost bandwidth.

RSVP is now supported by IBM's TCP/IP Version 4.1. For more information on RSVP, please consult RFC 2205.

10.5.3 Communications Server

Although TCP/IP has achieved tremendous popularity, it isn't necessarily the "best" protocol for all purposes. Other network protocols have particular

strengths and weaknesses. You can take advantage of the strengths of other protocols while still preserving the universal reach of TCP/IP on your network.

For example, IBM's SNA protocols work particularly well over WAN connections. If your network consists of remote offices, perhaps linked together with low bandwidth connections, then you may be able to improve the performance of TCP/IP applications by encapsulating that traffic in SNA protocols across the WAN. Benchmark results indicate that you can get more TCP/IP traffic across low bandwidth LAN connections if it's "wrapped" into SNA at one end and "unwrapped" at the other, particularly if you take advantage of SNA's end-to-end data compression.

IBM's Communications Server, available for a variety of platforms from PCs (OS/2 Warp, Windows NT) to mainframes, can be placed at both ends of a WAN connection to encapsulate TCP/IP into SNA.

For more information on IBM Communications Server, please visit:

<http://www.software.ibm.com/enetwork/commserver>

10.6 Related Publications

- *DNS and BIND*, 2nd Edition; O'Reilly & Associates; Albitz and Liu; ISBN: 1-56592-236-0.
Note: A third edition of this book will be available in September 1998.
- *IBM TCP/IP Performance Tuning Guide*, IBM, 1994, SC31-7188-0.
- *Inside TCP/IP Third Edition*, New Riders, 1997, 1-56205-74-6. Siyan.
- *Mastering TCP/IP for NT Server*, Sybex, 1997, 0-7821-2123-3. Minasi, Lammle and Lammle.
- *Learning Practical TCP/IP for AIX v3.2/v4.1 Users: Hints and Tips for Debugging and Tuning*, IBM, 1996, SG24-4381. Akihisa.
- RFC 1034 Domain Names - Concepts and Facilities.
- RFC 1035 Domain Names - Implementation and Specification.
- *TCP/IP Architecture, Protocols and Implementation with IP v6 and IP Security*, McGraw-Hill, 1996, 0-07-021389-5. Feit.
- *TCP/IP Tutorial and Technical Overview Fifth Edition*, IBM, 1995, GG24-3376. Murphy, Enders and Hayes.

also

- `man` pages on most UNIX systems for `nslookup`, `resolver`

Chapter 11. Managing Your Network

This chapter provides insight into how to respond to changes in your network quickly and easily. Read how to make changes to your TCP/IP network from any location, adding or deleting new users and connections as needed. Plus, we'll reveal how to generate reports which show how your network is being used so you can keep close tabs on its performance and security.

11.1 Remote Administration of DHCP and DDNS

Remote administration of the DHCP and DDNS servers can be accomplished using more than one method, and from more than one type of client platform.

One method of remote administration can be accomplished without adding any extra software beyond what is delivered with OS/2 Warp Server (with TCP/IP Version 4.1 for OS/2) and a OS/2 Warp 4 client. Another method adds a Web server to the OS/2 Warp Server platform. We used Domino Go Webserver, but any Web server that supports mapping should work. Note that Domino Go Webserver is available for free download. If you want to use the Secure Sockets Layer Security, you will have to register the product, but this is not mandatory. We will describe both methods of setting up remote administration.

A bit of explanation is in order first, though. The administration programs for both DHCP and DDNS are supplied as Java programs. The programs are compiled so that they can run as stand-alone programs or as applets from within a Java-enabled browser, or using an applet viewer.

11.1.1 Look! No Extra Software

Without a Web server on the OS/2 Warp Server platform, you can still remotely administer the DHCP/DDNS server by invoking similar scripts to those invoked locally on the server. The difference in the scripts is that the paths used have to reflect the network drive as seen from the client.

We have included some modified command files that will run on OS/2 or Windows 95/NT with REXX. REXX is shipped with OS/2 and is available for Windows platforms. You can also follow the instructions included with the TCP/IP product:

1. In the TCP/IP Information folder, double-click on **TCP/IP Guide**.
2. Search for JAVA and choose **Installing the TCP/IP JAVA Configuration Applications on a Remote Workstation**.

We tested those instructions on Windows 95, and they worked well for us. The command files that run on the server (and are mentioned in the online documentation) are as follows. Our rewritten scripts add a capital 'R' to the front of the script names. Note that in the case of DHCPSCPC.CMD the new name is RDHCPSCP.CMD (the final 'C' is dropped in the name to keep an 8.3 file name).

- DHCP Server Administration

- DDNSAPC.CMD

- It represents the main GUI to configure the DHCP server.

- DHCP Server Administration

- DADMGUI.CMD

- It shows DHCP server statistics and provides a function to reinitialize the DHCP server.

- DDNS Server Administration

- DHCPSCPC.CMD

- It represents the actual DDNS server configuration.

Here are the rewritten scripts that allow remote administration from an OS/2 workstation using Netscape 2.02 and Java 1.1.4 (including fixes). If you are using Netscape 2.02 with Java 1.1.6 you don't need to turn off the Just In Time compiler (JIT), which can be done by setting the `JAVA_COMPILER` environment variable to any value. See 11.1.2, "Using a Web Server" on page 375 for more information on Java-version-specific dependencies.

The server drive that contains the following scripts would get mapped as a network drive for the client. The client would execute the rewritten scripts from that network drive.

Note: Lines that are longer than 80 characters, such as `SET CLASSPATH`, are wrapped to the next line.

RDDNSAPC.CMD is the rewritten DDNSAPC.CMD.


```

/*
** file   : Rddnsapc.cmd
**         Remote DHCP Server Configuration
** purpose: to invoke DHCP Server Configuration from a client machine that
**         has a network drive mapped to the TCPIP drive of the server.
**         This script must be located on the server drive with \tcip installed.
**         This script will automatically pick off the drive letter and
**         use it correctly.
**
** CAVEATS:
** 1) Client machine must have REXX installed. REXX is available for
**    Windows 95/NT also (see http://rexx.hursley.ibm.com/rexx/rexxibm.htm)
**
** history:
** 02 May 1998 added command line argument for server name to configure, m3m
** 27 April 1998
** modified from ddnsapc.cmd by Michael McDaniel (m3m), The Fourth Crusade
**
**
*/
call RxFuncAdd 'SysLoadFuncs', 'RexxUtil', 'SysLoadFuncs'
call SysLoadFuncs

/* trace ?i */

parse arg server

if server == '' then
do
  parse source os . cmd
  say
  say 'usage:' cmd 'servername to configure'
  say
  rc = beep(750,750);
  EXIT
end

parse source os . cmd
JD = substr(cmd,1,1) /* JD for Java Drive */

signal on syntax name badsyntax
signal on halt name halt

original_dir = directory();
rc = directory(JD':\tcip\java');

'SET JAVA_COMPILER=XYZ;'

'SET
CLASSPATH='JD':\TCPIP\java\ddnssgui.jar;'JD':\TCPIP\java\tcpauth.jar;'JD':\TCPIP\jav
a\netdiver.jar;'JD':\TCPIP\java\jvc.jar;'JD':\TCPIP\java%\tcplang%\ddnsres.jar;'JD':
\TCPIP\java%\tcplang%\tcpares.jar;'
'start /c/min java COM.ibm.raleigh.ddnssgui.client.DDNSAdministratorClient
%tcplang%' server

rc = directory(original_dir);

EXIT

```

Figure 243. RDDNSAPC.CMD (Part 1 of 2)

```

/*****
halt:
    rc = directory(original_dir);
    say 'halt occurred at line' sigl
EXIT

badsyntax:
    say 'bad syntax at line' sigl
    rc = directory(original_dir);
return rc
*****/
/* end Rddnsapc.cmd */

```

Figure 244. RDDNSAPC.CMD (Part 2 of 2)

RDADMGUI.CMD is the rewritten DADMGUI.CMD.

```

/*
** file   : Rdadmgui.cmd
**
**          Remote DHCP Server Administration
** purpose: to invoke DHCP Server Administration from a client machine that
**           has a network drive mapped to the TCPIP drive of the server.
**           This script must be located on the server drive with \tcpip installed.
**           This script will automatically pick off the drive letter and
**           use it correctly.
**
** CAVEATS:
** 1) Client machine must have REXX installed. REXX is available for
**    Windows 95/NT also (see http://rexx.hursley.ibm.com/rexx/rexxibm.htm)
**
** history:
** 27 April 1998
** modified from dadmgui.cmd by Michael McDaniel, The Fourth Crusade
**
**
**
*/

call RxFuncAdd 'SysLoadFuncs', 'RexxUtil', 'SysLoadFuncs'
call SysLoadFuncs

/* trace ?i */

parse source os . cmd
JD = substr(cmd,1,1) /* JD for Java Drive */

signal on syntax name badsyntax
signal on halt name halt

original_dir = directory(JD':\tcpip\java');

```

Figure 245. RDADMGUI.CMD (Part 1 of 2)

```

'SET JAVA_COMPILER=xxx;'
'SET
CLASSPATH='JD':\TCPIP\java\dadmgui.jar;'JD':\TCPIP\java\tcpauth.jar;'JD':\TCPIP\java
\netdiver.jar;'JD':\TCPIP\java\%tcplang%\dadmres.jar;'JD':\TCPIP\java\%tcplang%\tcpa
res.jar;'
'start /c/min java COM.ibm.raleigh.dadmgui.DHCPadmin LANG=%tcplang%'

rc = directory(original_dir);

EXIT

/*****/

halt:
    rc = directory(original_dir);
    say 'halt occurred at line' sigl
EXIT

badsyntax:
    say 'bad syntax at line' sigl
    rc = directory(original_dir);
    return rc

/*****/

/* end Rdadmgui.cmd */

```

Figure 246. RDADMGUI.CMD (Part 2 of 2)

RDHCPSCP.CMD is the rewritten DHCPSCPC.CMD file.

```

/*
** file   : Rdhcpscp.cmd
**         Remote DDNS Server Configuration
** purpose: to invoke DHCP Server Configuration from a client machine that
**         has a network drive mapped to the TCPIP drive of the server.
**         This script must be located on the server drive with \tcpip installed.
**         This script will automatically pick off the drive letter and
**         use it correctly.
**
** CAVEATS:
** 1) Client machine must have REXX installed. REXX is available for
**    Windows 95/NT also (see http://rexx.hursley.ibm.com/rexx/rexxibm.htm)
**
** history:
** 02 May 1998 added command line argument for server name to configure, m3m
** 27 April 1998
** modified from dhcpscpc.cmd by Michael McDaniel, The Fourth Crusade
**      ^^^^^^^
**      note the last 'c' was dropped to keep 8 char file name
**
**
*/

call RxFuncAdd 'SysLoadFuncs', 'RexxUtil', 'SysLoadFuncs'
call SysLoadFuncs

/* trace ?i */

parse arg server
if server == '' then
do
  parse source os . cmd
  say
  say 'usage:' cmd 'servername to configure'
  say
  rc = beep(750,750);
  EXIT
end
parse source os . cmd
JD = substr(cmd,1,1) /* JD for Java Drive */

signal on syntax name badsyntax
signal on halt name halt

original_dir = directory();
rc = directory(JD':\tcpip\java');

'SET JAVA_COMPILER=XYZ;'

'SET
CLASSPATH='JD':\TCPIP\java\dhcpsgui.jar;'JD':\TCPIP\java\ddnssgui.jar;'JD':\TCPIP\ja
va\tcpauth.jar;'JD':\TCPIP\java\netdiver.jar;'JD':\TCPIP\java\jvc.jar;'JD':\TCPIP\ja
va\%tcplang%\dhcpress.jar;'JD':\TCPIP\java\%tcplang%\tcpares.jar;'

```

Figure 247. RDHCPSCP.CMD (Part 1 of 2)

```

'start /c/min java COM.ibm.raleigh.dhcpsgui.view.DHCPServerGUI %tcplang%' serverrc =
directory(original_dir);

EXIT

/*****/

halt:
    rc = directory(original_dir);
    say 'halt occurred at line' sigl
EXIT

badsyntax:
    say 'bad syntax at line' sigl
    rc = directory(original_dir);
    return rc

/*****/

/* end Rdhcpscp.cmd */

```

Figure 248. RDHCPSCF.CMD (Part 2 of 2)

11.1.2 Using a Web Server

We successfully remotely administered our DHCP/DDNS servers from multiple platforms using multiple tools. We used Java 1.1.x-enabled browsers and the applet viewer included with Java 1.1.4 for OS/2, also included with Sun JDK 1.1.5 for Windows 95, NT, and Sun JDK 1.1.3 on a Sun UltraSparc workstation.

Note: We strongly recommend using Java 1.1.6 for OS/2, available from IBM's Software Choice Web site, rather than the 1.1.4 version. This ensures the most current version of Java being used and, in addition, it avoids installing Java FixPaks.

We used Sun's HotJava browser on each of these platforms as well to successfully remotely administer our servers. We used Netscape V2.02 on OS/2 and NetScape V4.0 on Windows 95 and NT.

This section tells you how to set up the server to enable remote management using either a browser or the applet viewer included with Sun's JDKs.

Always remember to download the latest browser code available and to use the latest JDKs available (assuming, of course, that you are not back-level for some other specific reason).

At the time this publication was written, the following NetScape for OS/2 browser configurations were possible:

- NetScape V2.02 (the latest revision from Software Choice) with Java 1.02 (The Java version that comes with OS/2 Warp 4).

This configuration does not support remote configuration of TCP/IP Version 4.1 for OS/2.

- NetScape V2.02 (the latest revision from Software Choice) with Java 1.1.4 (along with the latest fixes for Java 1.1.4).

We recommend turning off the Just In Time (JIT) compiler. You can do this by setting the environment variable, `JAVA_COMPILER`, to a nonsense value such as `XYZ` either in the `CONFIG.SYS` file or temporarily from an OS/2 command line by typing the following commands:

```
SET JAVA_COMPILER=XYZ
NETSCAPE
```

This ensures Netscape will recognize that the JIT is turned off. If using Java 1.1.4 for OS/2, you can obtain the latest fixes from the following FTP site:

<ftp://ftp.hursley.ibm.com/pub/java/fixes/os2/11/114>

- NetScape V2.02 (the latest revision from Software Choice) with Java 1.1.6.

This configuration ensures best results. You don't need to turn off the Just In Time compiler.

On the server, you will need an HTTP server, such as Domino Go Webserver 4.6.2.5 for OS/2. To obtain a copy, see:

<http://www.software.ibm.com/webserver/dgw/prodlist.htm>

By default, the following two scripts will be running when you start your DDNS/DHCP server:

```
DDNSAPS.CMD
DHCPSCPS.CMD
```

Both files reside in the `\TCPIP\BIN` directory. These are the scripts that allow remote configuration. Actually, if you can locally start the DDNS and DHCP configuration, then these two scripts are already running. When the respective script for the DDNS or DHCP servers is not running and you try (locally) to start the configuration program, you receive the following error:

Communication could not be established with the server.

11.1.2.1 Domino Go Webserver for OS/2

The configuration file for the Domino Go Webserver is `\MPTN\ETC\HTTPD.CNF`, and we suggest that you back up the original file before doing any configuration. You can modify the file directly, or you can start your browser and use the GUI to configure Domino Go Webserver. Using the browser has the advantage of proving that the Domino Go Webserver is also running. At the server, enter the following Webserver URL:

`http://localhost`

You should get the Configuration and Administration Forms page. If not, enter the following Webserver URL to see if that works:

`http://localhost/admin-bin/cfgin/initial`

If you are remote, replace `localhost` by the fully qualified domain name (FQDN) of the HTTP server.

You will be prompted to enter user ID and password you defined when you installed the Domino Go Webserver.

Presumably, you have the Configuration and Administration page now. Here is how to configure Domino Go Webserver for DDNS and DHCP Remote Administration.

In our example, the following URL brings you to the Configuration and Administration Forms:

`http://itscfs00.itsc.austin.ibm.com/admin`

where `itscfs00.itsc.austin.ibm.com` represents the fully qualified domain name. To configure Domino Go Webserver to support remote configuration of TCP/IP Version 4.1 for OS/2, perform the following steps:

1. Make sure you are in the **Configuration and Administration Forms** page, as shown in Figure 249 on page 378.

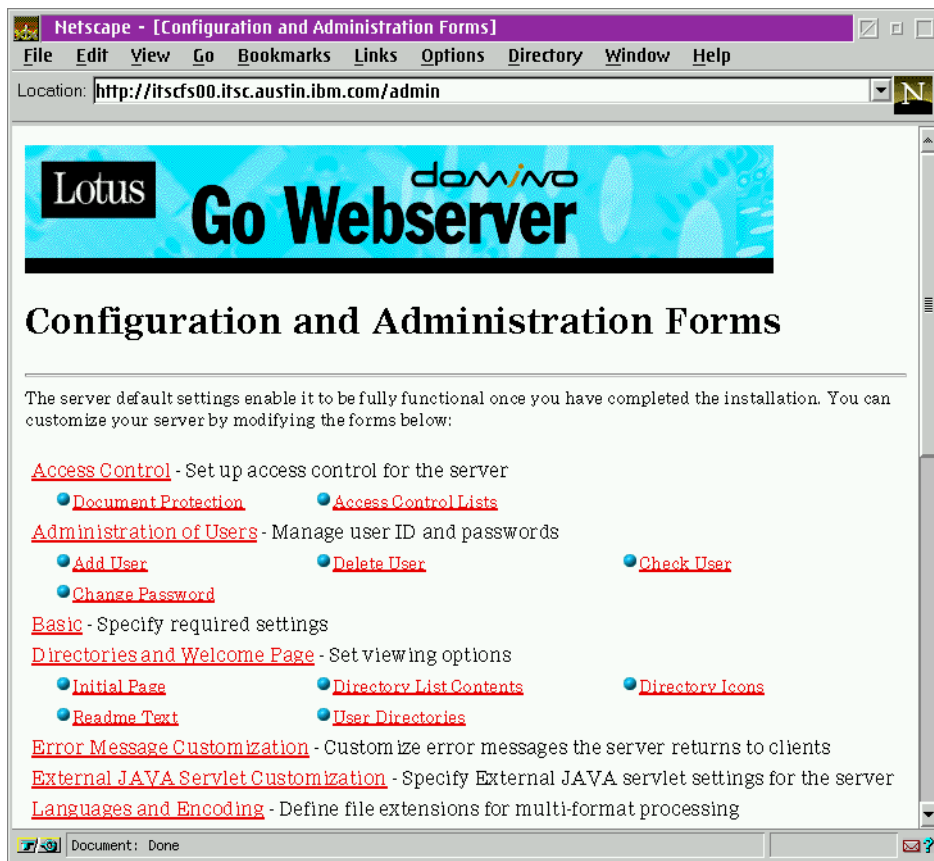


Figure 249. [Go Webserver] Configuration and Administration Forms

2. Scroll down to Request processing.
3. Click on **Request Routing** to display the Request Routing page as shown in Figure 250 on page 379.

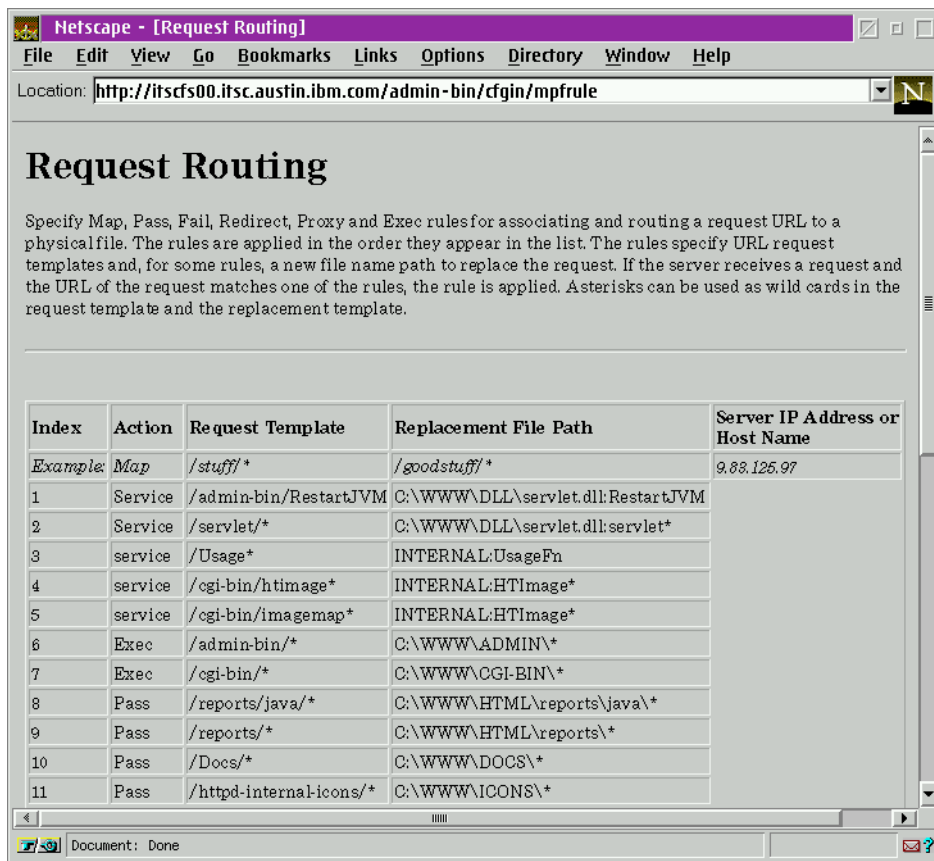


Figure 250. [Go Webserver] Request Routing Page

- As shown in Figure 251 on page 380, select the highest index number displayed (probably 15), select the **Insert before** radio button, and then change the Action to **Pass**. In the URL request template field, enter `/TCP/IP/*`. In the Replacement file path field, enter `C:\TCP/IP\JAVA*`, assuming C: is the drive where TCP/IP is installed on the server.

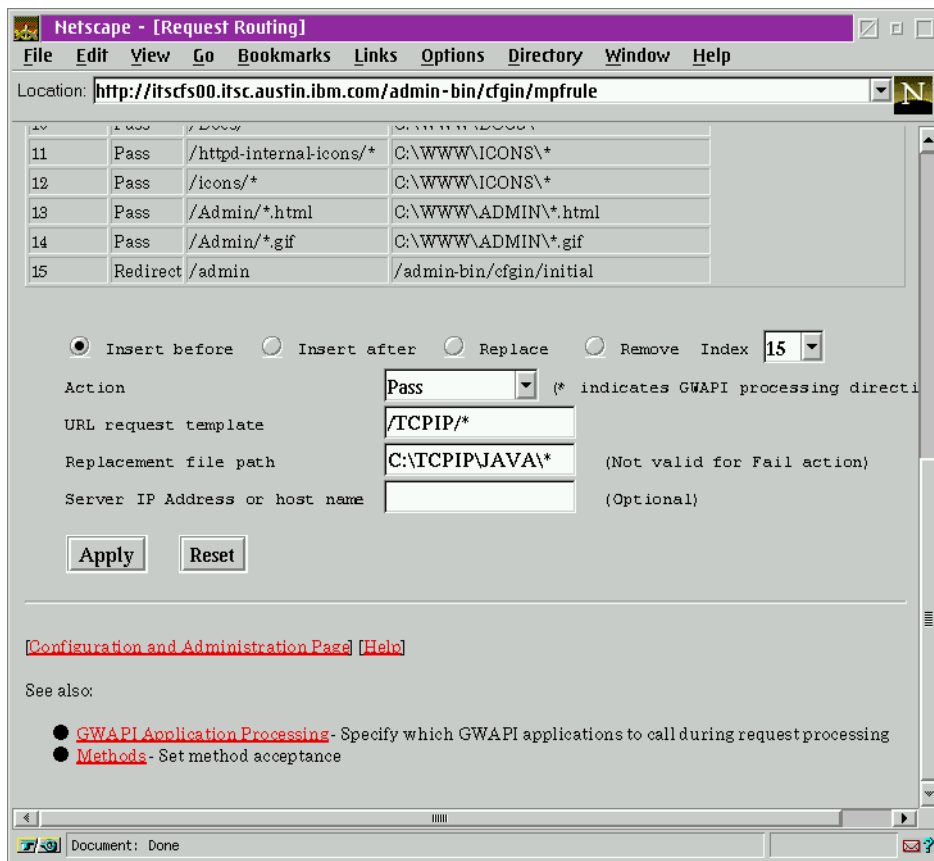


Figure 251. [Go Webserver] Insert Routing Request

- Click on **Apply**, and then when the Confirmation page appears, click on **Restart Server**. You should get a Restart Confirmation screen at this point.

The following entry was inserted in the Routing Request table:

Table 20. Routing Request Table Entry

Index	Action	Request Template	Replacement File Path
15	Pass	/TCPIP/*	C:\TCPIP\JAVA*

- On your server, in the C:\TCPIP\JAVA\EN_US directory, copy the TCPLOGIN.HTM file to WELCOME.HTM.

7. You may now test the setup by executing the following command from an OS/2 command prompt:

```
appletviewer http://localhost/tcpip/en_US/ddnsap.htm
```

Note: If not executed at the server, `localhost` should be replaced by the fully qualified domain name.

Alternatively, as shown in Figure 252, use the OS/2 Netscape browser with Java 1.1.6 for OS/2 and load the following URL:

```
http://itscfs00.itsc.austin.ibm.com/tcpip/en_US/welcome.htm
```



Figure 252. [OS/2 Warp] Remote Configuration of TCP/IP Version 4.1 for OS/2

where `itscfs00` is the hostname and `itsc.austin.ibm.com` the domain name where Domino Go Webserver is installed.

A very reliable way we have found to run the remote applets is using an applet viewer. The applet viewer is included with the Sun Java Development Kits for various platforms. The invocation line from a client follows for each portion of the management. Note that we copy our 'appletviewer' binary file over to an 'applet' binary file so we can type less. On OS/2 and Microsoft platforms, just copy `APPLETVIEWER.EXE` to `APPLET.EXE` in the JDK bin directory.

- DHCP Server Configuration

- `APPLET http://<hostname>.<domain>/tcpip/java/en_US/ddnsap.htm`

This command loads the main GUI to configure the server.

- DHCP Server Administration

- `APPLET http://<hostname>.<domain>/tcpip/java/en_US/dadmgui.htm`

This command shows server statistics and may also be used to reinitialize server.

- DDNS Server Administration

- `APPLET http://<hostname>.<domain>/tcpip/java/en_US/dhcpscp.htm`

This command loads the DHCP server configuration program (beats us why it's called "Administration").

Now you know what we found to be another very reliable method of remote invocation if you have your management Web server configured. For browser-based management, we used Sun's HotJava Version 1.1.2 browser. We found this to be more reliable than NetScape browsers (Windows versions; OS/2 version with Java 1.1.6 ok) or Internet Explorer (various versions). We tested the HotJava browser on OS/2 Warp 4 using JDK 1.1.4 (including fixes) and also on Windows 95 and NT using Sun's JDK1.1.5 for Windows 95/NT. The other browsers were tested on OS/2 and Windows 95 and NT. The HotJava browser proved to be robust on both OS/2 and Windows platforms.

The good news is that once the Java management applets download to your workstation, the management application runs very fast. The management applets are quite large, though, and take some time to download. Unfortunately, the HotJava browser does not provide feedback as to how far into the download you are; so patience is necessary. Once the applets download, they communicate directly with applications on the server and bypass the Web server. The applications themselves do not have a 'busy'

indication; so you will just have to wait when the Java client applications are communicating with the server applications.

The times you can expect for applet downloads are included below. It did not seem to matter whether the applet was invoked from a browser or using the applet viewer application; the download time was comparable for the same platform. Download speeds were comparable for the various platforms. Remember that these times were achieved on a 16 Megabits/second token-ring network. Scale your own times accordingly.

dhcpscp.htm - 17 seconds to get to the Start button, 45 seconds to get the initial notebook after you enter your password.

dadmgui.htm - 11 seconds to get to the Start button, 2 seconds to get to the main screen after you enter your password

ddnsap.htm - 24 seconds to get to the Start button, 15 seconds to get to the initial notebook screen after you enter your password

11.1.2.2 Lotus Domino

If you want to do remote administration from a Lotus Domino server, it is just as easy as from a Domino Go Webserver. You can have a Lotus Domino server on the same machine as your DHCP/DDNS or on a different machine. If you have the option, we suggest you run your Domino server on a different machine for performance reasons. If you set up like this, you will still need a Web server on the DHCP/DDNS server to serve up the HTML pages and Java applets. The Domino page would just be redirected to the other Web server. This configuration might occur if you already had an externally accessible Domino server and you wanted to add remote administration for a new DHCP/DDNS server you were configuring. In either case, the Domino configuration is similar, and we will cover both. We will start with a newly installed Domino server and explain how to set up the needed Domino database. If you already have a Web administration database configured, you can skip down a few steps.

1. Start Domino Server Administration.
2. Click on [**File — Database — New**].

The server is Local; the title does not matter (we suggest Domino Web Server Configuration), but the File Name must be DOMCFG.NSF, and highlight Domino Web Server Configuration for your template in the lower half of the screen. Note that the file name will not be correct unless you type it in yourself.

3. Click **OK**.

4. The next page that will come up is an About Domino Web Server Configuration. After reading it, just press **Escape** to get to the actual configuration page. You may want to slide the vertical bar to the right some so you can read the configuration choices easier.
5. If your DHCP/DDNS servers are on the same machine, you will map your URL to a directory. If your DHCP/DDNS servers are on a different machine with a Web server installed, you will redirect your URL to another URL. Follow steps 4 to 5, or 9 to 10 as the case may be. Both sets of instructions start at the Domino Web Server Configuration page which you can get to from the start up configuration page by clicking on **[File — Database — Open]** and choosing the database.
6. Click on **[Create — Mapping URL -> Directory]**. Skip the IP Address; fill in your chosen comment and use the following information:
URL path: /TCPIP/*
Directory: C:\TCPIP\JAVA
assuming TCP/IP is installed on the C: drive. Check that the Read radio button is set for **Access**.
7. Click **[File — Save]** and then **[File — Close]** to get back to the main administration screen.
8. Restart your Lotus Domino HTTP server (enter `HTTP quit`, load `HTTP`) or start Domino, as the case may be. Test by entering the following address in a Web browser:
`http://server.domain/tcpip/en_US/tcplogin.htm`
You should see the TCP/IP Administration page.
9. Click on **[Create — Redirection URL -> URL]**
10. Skip the IP Address; add your comment and fill in the following:
URL path: `http://thisserver.domain/tcpadmin`
Redirection URL: `http://otherserver.domain/tcpip/en_US/tcplogin.htm`

This presumes you set up the other DHCP/DDNS Web server as described in Part 11.1.2, "Using a Web Server" on page 375.

Appendix A. Creating DHCP Boot Diskettes

This appendix examines how to create bootable DOS and OS/2 Warp diskettes which work with your dynamic TCP/IP network, so you can install Windows, or just about any other kind of software, across the network to new PCs.

A.1 OS/2 Warp Boot Diskettes

Using just three diskettes, you can start almost any PC with OS/2 Warp 4, attach to the network using either a TCP/IP file system or FTP, and start copying files to install OS/2 Warp 4 or other OS/2 Warp software to a new PC. With this technique, you can quickly and easily partition, reformat, and load hard disks with OS/2 Warp software across a Dynamic IP network.

This section describes how to create a set of boot diskettes to attach to the network using DHCP. The major challenge with OS/2 Warp 4 is to find a way to fit enough advanced TCP/IP networking software on just three diskettes. Using some compression tricks, a virtual disk, and a bit of ingenuity, you can design your own set to work with just about any PC while still preserving the advantages of a Dynamic IP network where addresses are available on-the-fly.

A.1.1 Prerequisites

Before starting, you should have the following resources at your disposal:

1. Four blank, formatted diskettes (1.44 MB 3.5 inch).
2. Original IBM OS/2 Warp 4 CD-ROM.
3. Installed OS/2 Warp 4 system with CD-ROM drive and diskette drive, where you can prepare the boot diskettes.
4. Info-Zip's Zip and UnZip utilities, or suitable equivalents, available on the Internet at:

`ftp://ftp-os2.nmsu.edu/pub/os2/util/archiver`

5. If planning to use FTP to transfer files, a simple OS/2 Warp FTP utility such as GET105.ZIP, written by Keith Cotroneo, available on the Internet at:

`ftp://ftp-os2.nmsu.edu/pub/os2/apps/internet/ftp/client`

6. If not planning to use FTP, another file system, such as the IBM NFS Kit for OS/2 Warp TCP/IP (IBM Part No. 65G1255) or SRVIFS-IP by Sam Detweiler (included on the CD-ROM accompanying this book).

7. Server(s) able to provide appropriate file system and Dynamic IP capabilities on the network, such as an FTP server.
8. A client PC attached to the network and available for testing.
9. If not provided with OS/2 Warp 4, driver files for the client network adapter, including a correct NIF file.

A.1.2 Step-by-Step Procedure

We assume that you have the OS/2 Warp 4 CD-ROM available in Drive X. We also assume that Drive A contains the diskettes you will be working on. Substitute the correct drive letters in the following steps if necessary.

1. Create basic OS/2 Warp 4 bootable diskettes. A utility called SEDISK is provided on the OS/2 Warp 4 CD-ROM to help prepare diskettes appropriate to the particular type of system you plan to use. From the OS/2 Warp command line, type:

```
X:\CID\EXE\OS2\SEDISK /S:X:\OS2IMAGE /T:A: /P:36
```

The last part ("/P:36") is optional. Our test system was an IBM ThinkPad 760ED, and "/P:36" adds PCMCIA support to the diskettes for this particular notebook computer. The text file X:\SAMPLE.RSP lists other numeric codes available for many other notebook computers.

SEDISK will require three of your four diskettes. Label the first "Installation Diskette (Diskette 0)," the second "Diskette 1," and the third "Diskette 2." Set these diskettes aside.

2. Create a diskette with basic TCP/IP networking. A utility called THINLAPS is provided on the OS/2 Warp 4 CD-ROM to help add files necessary to attach to a network. From the OS/2 Warp command line, type:

```
X:\CID\IMG\MPTS\THINLAPS X:\CID\IMG\MPTS A: IBMTOKCS.NIF /TCP/IP /DHCP
```

IBMTOKCS.NIF is the name of the NIF file for the IBM PCMCIA Token-Ring Adapter, the adapter installed in our ThinkPad 760ED. You should substitute the name of the NIF file for your network adapter. (The NIF files are installed in the \IBMCOM\MACS directory on your OS/2 Warp 4 system. You can view these with any text editor if you wish. Consult the online Network Adapters and Protocol Services Guide if you wish to use a NIF file and driver which does not ship with OS/2 Warp 4.)

THINLAPS will ask you to reinsert Diskette 1 in order to modify CONFIG.SYS as required. Label the fourth diskette "Diskette 3 - Network Files" and set it aside.

3. Edit CONFIG.SYS on Diskette 1. Use TEDIT to make some changes to CONFIG.SYS. (You may wish to save a copy of the original in case you

make a mistake.) Our CONFIG.SYS for the ThinkPad 760ED follows, and changes are highlighted.

```
buffers=32
iopl=yes
memman=swap,delayswap
protshell=CMD.EXE /K A:\STARTUP.CMD
set os2_shell=cmd.exe
diskcache=64
protectonly=yes
libpath=.;\;
ifs=hpfs.ifs /c:64
pauseonerror=no
codepage=850
devinfo=kbd,us,keyboard.dcp
devinfo=scr,ega,vt1850.dcp
device=\dos.sys
REM device=\mouse.sys
set path=\;Z:\CMD;Z:\EXE;Z:\DLL;
set dpath=\;Z:\CMD;Z:\EXE;
set keys=on
basedev=ibmkbd.sys
basedev=ibm1flpy.add
basedev=ibm1s506.add
basedev=ibm2flpy.add
basedev=ibm2adsk.add
basedev=ibm2scsi.add
basedev=ibmint13.i13
basedev=os2dasd.dmd
device=\testcfg.sys
basedev=xdfloppy.flt
REM device=\refpart.sys
REM device=\pmdd.sys
REM set video_devices=vio_vga
REM set vio_vga=device(bvhvga)
REM set copyfromfloppy=1
BASEDEV=PCMCIA.SYS
BASEDEV=IBM2SS04.SYS
BASEDEV=IBM2SS14.SYS
DEVICE=\VDISK.SYS 2000,,
SET SAVECONNECT=1
rem *** Start of ThinLAPS additions ***
call=netbind.exe
run=lanmsgex.exe
device=lanmsgdd.os2
device=protman.os2 /I:a:\
device=sockets.sys
```

```

device=afinet.sys
device=ifndis.sys
device=IBMTOKCS.OS2
rem *** End of ThinLAPS additions ***
IFS=SRVIFS2.IFS
SET DIRCMD=/O:GN /N /V

```

Changes to CONFIG.SYS include:

- PROTSHELL was changed to run STARTUP.CMD, a batch file, used to copy files, attach to the network, and automatically start the tasks that you wish to perform.
- DISKCACHE, the disk cache for the FAT file system, was lowered to 64 to conserve memory. You can adjust the FAT and HPFS disk caches according to your particular performance needs and memory constraints.
- Certain directories in LIBPATH were removed to prevent potential access to DLLs in unknown locations.
- DEVINFO was changed to EGA, primarily to conserve disk space. Since only a text display is available from a diskette boot, and since all modern PCs are still compatible with EGA, it makes little sense to provide space consuming VGA display services.
- MOUSE.SYS is not needed since mouse support is not required with a simple text screen.
- PATH and DPATH have been modified to provide access to Drive Z, a TCP/IP network drive provided by NFS or SRVIFS-IP. Feel free to customize these lines depending on your needs.
- REFPART.SYS and PMDD.SYS device drivers are omitted, again to conserve disk space. The services provided by these device drivers are not required by most systems.
- VGA support statements have been REMmed out as well, again to conserve disk space.
- SET COPYFROMFLOPPY is REMmed out to prevent OS/2 Warp's installation program from copying files from these diskettes.
- PCMCIA.SYS, IBM2SS04.SYS, and IBM2SS14.SYS are loaded to provide PCMCIA support for our ThinkPad 760ED. These lines may be different in your CONFIG.SYS (or may not exist at all for a desktop PC). We removed all parameters from these lines to avoid some error messages.
- VDISK.SYS is loaded to provide a 2 MB ("2000") virtual RAM disk. This RAM disk provides space for loading DLLs, programs, and other files

from compressed form on the diskette. It's needed because we cannot expect hard disk or network drive space during the boot process. Note that VDISK.SYS uses the file OSO001.MSG in order to display messages during bootup. We do not have the space available on the diskettes to hold OSO001.MSG, so an error message is displayed during bootup. However, VDISK still loads correctly.

- Some OS/2 Warp installation routines expect SET SAVECONNECT=1 in order to provide a network connection after bootup.
 - The section of "ThinLAPS additions" in CONFIG.SYS has been modified slightly so that the connection to the TCP/IP network does not start right away. That process has been moved to a batch file called SETUP.CMD, which is called from the STARTUP.CMD batch file. IBMTOKCS.OS2 is the device driver used for the IBM PCMCIA Token-Ring Adapter, and this line will be different for other network adapters. Some releases of MPTS (Multiprotocol Transport Services) include AFLEAN.SYS. AFLEAN.SYS may be used instead of AFINET.SYS in order to save space on the diskette.
 - SRVIFS2.IFS, part of SRVIFS-IP, is loaded to provide a TCP/IP file system. You may elect to use NFS instead or omit this line entirely if using FTP.
 - The SET DIRCMD line is entirely optional. It's used here so that a directory listing (DIR) is sorted alphabetically and formatted differently.
4. Delete unnecessary files on Diskette 2 and add necessary files from Diskette 3. Diskette 2 (the third diskette) is the last diskette actually used in the boot process. All the required networking files from Diskette 3, created by THINLAPS, need to be squeezed onto Diskette 2.

In order to properly "merge" these diskettes, try copying all the files on Diskettes 2 and 3 to a temporary directory on your hard disk. Delete, modify, and move the files in this working directory as needed. Here's a listing of the files that should be on Diskette 2 when you're done:

AFINET	SYS	225479	8-11-96	10:05p
BKSCALLS	DLL	512	8-12-96	3:00a
BMSCALLS	DLL	512	8-12-96	3:02a
BVHINIT	DLL	10540	8-12-96	2:08a
BVSCALLS	DLL	512	8-12-96	2:59a
CMD	EXE	74640	8-09-96	12:15a
COUNTRY	SYS	36185	8-09-96	12:29a
DOS	SYS	1142	12-04-95	11:22p
DOSCALL1	DLL	123680	8-27-96	8:45a
FILES	EXE	482698	4-03-98	11:22a
HPFS	IFS	141378	8-13-96	11:17a

IBMTOKCS	OS2	28724	4-30-96	2:06p
IFNDIS	SYS	35828	8-11-96	10:04p
KBDCALLS	DLL	1024	8-12-96	3:02a
KEYBOARD	DCP	28097	8-09-96	12:58a
LANMSGDD	OS2	3604	8-01-96	6:14p
LANMSGDL	DLL	2580	8-01-96	6:14p
LANMSGEX	EXE	1099	8-01-96	6:14p
LT0	MSG	12943	8-01-96	6:14p
LTG	MSG	4487	3-28-96	10:11a
MSG	DLL	512	8-12-96	2:56a
NAMPIPES	DLL	1024	8-12-96	3:02a
NETBIND	EXE	13657	8-01-96	6:13p
NLS	DLL	512	8-12-96	2:55a
OS2CHAR	DLL	512	8-12-96	2:58a
PRO	MSG	2234	8-01-96	6:13p
PROTMAN	OS2	22308	8-01-96	6:13p
PROTOCOL	INI	412	4-02-98	5:15p
QUECALLS	DLL	1024	8-12-96	2:57a
SESMGR	DLL	1536	8-12-96	2:52a
SETUP	CMD	122	4-03-98	10:38a
SOCKETS	SYS	59565	8-15-96	3:47p
SRVIFS2	IFS	23087	1-06-93	9:39p
STARTUP	CMD	1805	4-03-98	10:13a
TESTCFG	SYS	9808	8-12-96	2:21a
VDISK	SYS	2904	8-12-96	2:24a
VIOCALLS	DLL	2048	8-14-96	5:06a
VTBL850	DCP	10478	8-09-96	1:00a

Some comments on particular files follow:

- IBMTOKCS.OS2 is the driver for the IBM PCMCIA Token-Ring Adapter. You may be using another driver.
- PROTOCOL.INI is a text file containing adapter and protocol settings. In a token-ring network, for example, you may wish to edit this file to change the default ring speed used by the IBM PCMCIA Token-Ring Adapter. Other network adapters will have different settings. THINLAPS attempts to create a default PROTOCOL.INI which works in most situations.
- SRVIFS2.IFS is the file system driver for SRVIFS-IP. If you are not using SRVIFS-IP, this file is not required.
- VDISK.SYS can be found in the \OS2\BOOT directory on your OS/2 Warp 4 system.

FILES.EXE, SETUP.CMD, and STARTUP.CMD require some additional work to set up properly.

FILES.EXE is a self-extracting, compressed Zip file, containing the files needed to attach to the network. (We created FILES.EXE using Info-Zip's Zip and Zip2Exe utilities.) These files are extracted to the RAM disk created by VDISK.SYS during the boot process. Many of the remaining files on Diskettes 2 and 3 are contained in FILES.EXE, but not all. For our ThinkPad 760ED, we included the following files in FILES.EXE:

Archive: FILES.EXE				
Length	Size	Ratio	Date	Name
-----	-----	-----	----	-----
512	205	60%	08-12-96	ANSICALL.DLL
19893	11872	40%	08-23-96	ARP.EXE
11587	6346	45%	08-23-96	CNTRL.EXE
168538	131892	22%	08-11-96	DHCPD.EXE
24710	19384	22%	08-11-96	DHCPSTRT.EXE
0	0	0%	04-02-98	ETC/
13218	3619	73%	08-11-96	ETC/DHCPD.CFG
5451	2079	62%	08-11-96	ETC/PROTOCOL
42725	9073	79%	08-11-96	ETC/SERVICES
55296	27572	50%	01-10-98	GET.EXE
9415	7731	18%	08-13-96	HARDERR.EXE
17193	10234	41%	08-23-96	IFCONFIG.EXE
24064	13180	45%	08-23-96	INETWAIT.EXE
21812	16318	25%	08-12-96	NPXEMPLTR.DLL
35552	19216	46%	08-23-96	PING.EXE
20671	12002	42%	08-23-96	ROUTE.EXE
37842	15905	58%	08-23-96	SO32DLL.DLL
9035	6266	31%	01-31-93	SRVATTCH.EXE
21947	12813	42%	06-07-93	SRVRDR.EXE
62289	30200	52%	08-23-96	TCP32DLL.DLL
47279	24151	49%	08-23-96	TCPIPDLL.DLL
34304	9344	73%	08-11-96	TCPMRI.DLL
32288	17551	46%	08-23-96	TCPTIME.DLL
9390	8330	11%	11-04-96	TEDIT.EXE
14596	3053	79%	11-04-96	TEDIT.HLP
-----	-----	---	-----	-----
739607	418336	43%		25 files

Some of these files, in particular, deserve comment:

- GET.EXE is a simple FTP program we chose among the several available on the Internet. It takes up very little space (just 27 KB compressed) and can be used in batch files.
- PING.EXE is optional (but useful). We extracted PING.EXE from X:\CID\IMG\MPTS\MPTN\BIN\BIN.ZIP.
- SRVATTCH.EXE and SRVRDR.EXE are part of SRVIFS-IP.

- TCPTIME.DLL is required by the version of INETWAIT.EXE in the original release of TCP/IP Version 4.0 for OS/2 Warp. We extracted TCPTIME.DLL from X:\CID\IMG\MPTS\MPTN\DLL\MDLL.ZIP.
 - TEDIT.EXE and TEDIT.HLP are both optional but quite useful. This simple text editor can be used to make changes to CONFIG.SYS, STARTUP.CMD, and/or SETUP.CMD when troubleshooting. You can find both these files in the \OS2 directory on your OS/2 Warp 4 system.
5. Create STARTUP.CMD and SETUP.CMD. STARTUP.CMD and SETUP.CMD are the key batch files used at bootup. Here's how our STARTUP.CMD file looks:

```
@echo off
set vdisk=r:
copy a:files.exe %vdisk% 1>nul 2>nul
if not errorlevel 1 goto config:
set vdisk=q:
copy a:files.exe %vdisk% 1>nul 2>nul
if not errorlevel 1 goto config:
set vdisk=p:
copy a:files.exe %vdisk% 1>nul 2>nul
if not errorlevel 1 goto config:
set vdisk=o:
copy a:files.exe %vdisk% 1>nul 2>nul
if not errorlevel 1 goto config:
set vdisk=n:
copy a:files.exe %vdisk% 1>nul 2>nul
if not errorlevel 1 goto config:
set vdisk=m:
copy a:files.exe %vdisk% 1>nul 2>nul
if not errorlevel 1 goto config:
set vdisk=l:
copy a:files.exe %vdisk% 1>nul 2>nul
if not errorlevel 1 goto config:
set vdisk=k:
copy a:files.exe %vdisk% 1>nul 2>nul
if not errorlevel 1 goto config:
set vdisk=i:
copy a:files.exe %vdisk% 1>nul 2>nul
if not errorlevel 1 goto config:
set vdisk=h:
copy a:files.exe %vdisk% 1>nul 2>nul
if not errorlevel 1 goto config:
set vdisk=g:
copy a:files.exe %vdisk% 1>nul 2>nul
if not errorlevel 1 goto config:
set vdisk=f:
```

```

copy a:files.exe %vdisk% 1>nul 2>nul
    if not errorlevel 1 goto config:
    set vdisk=e:
copy a:files.exe %vdisk% 1>nul 2>nul
    if not errorlevel 1 goto config:
    set vdisk=d:
copy a:files.exe %vdisk% 1>nul 2>nul
    if not errorlevel 1 goto config:
    set vdisk=c:
copy a:files.exe %vdisk% 1>nul 2>nul
    if not errorlevel 1 goto config:
    goto error
:config
    %vdisk%
    copy a:\*.dll
    copy a:\*.exe
    copy a:\*.cmd
    set path=%vdisk%\;%path%
    set dpath=%vdisk%\;%dpath%
    set beginlibpath=%vdisk%\;a:\;
    files
    del files.exe
    if not exist setup.cmd goto end:
    call setup
    goto end:
:error
    Echo Error finding VDISK
:end

```

The first several lines in this batch file attempt to locate the RAM disk created with VDISK.SYS, starting with Drive R and working back to Drive C. Once located, all the necessary files from Diskette 2 are copied to the RAM disk. Some environment variables are set (to allow FILES.EXE and other programs to run properly); then FILES.EXE is run to extract the compressed files inside. Finally, if SETUP.CMD exists, STARTUP.CMD calls it.

SETUP.CMD is considerably shorter and looks like this:

```

SET ETC=%vdisk%\ETC
DETACH CNTRL.EXE
INETWAIT
DHCPSTRT -i lan0
DETACH SRVRDR.EXE
SRVATTCH Z: \\192.168.6.12\TESTDRV

```

The first line sets the ETC environment variable, required by many TCP/IP applications. The ETC directory contains critical configuration files. Then

CNTRL.EXE is started as a background task. INETWAIT runs (to wait for the TCP/IP stack to initialize); then DHCP is started (to obtain address information and connect to the network over "lan0," in this case the IBM PCMCIA Token-Ring Adapter).

The last two lines start SRVIFS-IP and create a network drive (Drive Z) from 192.168.6.12, a SRVIFS-IP server. (If a name server is available, and its address is returned by the DHCP server, the IP name can be specified instead.) You may wish to use NFS, GET (FTP) or some other method to obtain additional files across the network. Any additional commands can be added to this batch file as needed.

6. Test and verify your network connection. We've included a few programs on these boot diskettes so you can troubleshoot your connection to the network. These files include:
 - LANTRAN.LOG will be created on Diskette 2 and will contain information on whether the network adapter was able to attach to the network. Use the information in this file to determine whether the network adapter driver is functioning correctly and whether PROTOCOL.INI is constructed properly.
 - ARP.EXE lets you look up an IP address based on a network adapter address.
 - IFCONFIG lets you determine the IP address assigned to the system. At the command line, after booting from these diskettes, type (in lowercase):

```
ifconfig lan0
```
 - PING.EXE allows you to test the connection to a particular system. It can be particularly helpful when you're installing both a new network and a new set of PCs. You can test to see whether traffic is being routed correctly.

A.1.3 Notes on the Step-by-Step Procedure

1. Try to use TEDIT when making changes to text files on the diskettes. The OS/2 System Editor will create extended attributes for each text file you edit, and these extended attributes take up extra space. Extended attributes are not required for boot diskettes.
2. Info-Zip's Zip utility can accept the -9 option, resulting in the highest possible degree of compression. Then the Zip2Exe utility can convert the Zip file to FILES.EXE. We recommend using -9 to conserve as much space as possible. You should also use -r to properly store the files in the ETC directory.

3. While we opted to use OS/2 Warp files obtained from the original CD-ROM release, you may wish to use files for these boot diskettes from OS/2 Warp FixPaks and later versions of MPTS and TCP/IP. If so, there may be dependencies on certain DLLs which we have not included. Newer is not necessarily better, however. Files tend to grow in size over time as features are added.
4. SNIFFLE, part of the MPTS supplemental utilities, can be used to verify the construction of a network adapter driver's NIF file. See the online Network Adapters and Protocol Services Guide for details.
5. See A.3, "Related Publications" on page 400, to get more information on software distribution techniques with OS/2 Warp.

A.2 DOS Boot Diskettes

With DOS boot diskettes that can attach to a file server, such as Warp Server or Windows NT, you can install many different software packages across the network, including versions of Windows with DOS-based installation programs.

This section describes how to create a pair of DOS boot diskettes which can provide a PC with a TCP/IP (TCPBEUI) or NetBEUI-based file system and a dynamically assigned IP address.

A.2.1 Prerequisites

These are the necessary prerequisites:

1. Two blank, formatted diskettes (1.44 MB 3.5 inch)
2. A workstation that has either BASIC or FULL DOS LAN Services (DLS) and IBM PC-DOS 2000 installed
3. Server(s) able to provide appropriate file system and Dynamic IP capabilities on the network, such as Warp Server
4. A client PC attached to the network and available for testing

A.2.2 Step-by-Step Procedure

We assume that you have DOS LAN Services installed on Drive C in the \NET subdirectory. We also assume that Drive A will contain any diskettes you will be working with. Substitute the correct drive letters in the following steps if necessary.

1. Create Diskette 1. Due to the size of DLS we have to split the software across two diskettes. Diskette 1 contains the files to boot the PC and

initialize the network adapter. It will also contain files which allow you to partition and format a hard drive, and to edit files. Label this diskette "DLS Boot Diskette 1." It should be formatted under IBM PC-DOS 2000 using the `FORMAT A: /S` command, to create a bootable diskette.

Then copy the following files from the \DOS subdirectory to Diskette 1:

- EMM386.EXE
- HIMEM.SYS
- SETVER.EXE
- SHARE.EXE
- FORMAT.COM
- FDISK.COM
- E.EXE
- E.EX
- E.INI

Copy the following files from the \NET subdirectory to the diskette:

- DLSHELP.SYS
- LT2.MSG
- NTSTS.DOS
- PROTMAN.DOS
- IBMTOK.DOS

Our test system was an IBM ValuePoint with an IBM Token-Ring 16/4 network adapter; so we copied the IBMTOK.DOS driver file. You may have to copy a different driver depending on the type of network adapter you have in your PC.

2. Create CONFIG.SYS. In order to boot a new PC and partition its hard drive without having to wait for a network connection to be established. We can create a menu-based CONFIG.SYS on Diskette 1 as follows:

```
[MENU]
MENUITEM=NETWORK, Connect to the network
MENUITEM=STANDALONE, No network connection
MENUDEFAULT=NETWORK,15

[COMMON]
FILES=30
BUFFERS=10
DOS=HIGH,UMB
DEVICE=A:\HIMEM.SYS
DEVICE=C:\DOS\EMM386.EXE noems ram x=d800-e2ff
DEVICEHIGH=C:\DOS\SETVER.EXE

[NETWORK]
```

```

DEVICEHIGH=A:\PROTMAN.DOS /i:A:\
DEVICEHIGH=A:\IBMTOK.DOS
DEVICEHIGH=A:\NTSTS.DOS
DEVICEHIGH=A:\DLSHELP.SYS
LASTDRIVE=Z

```

```
[STANDALONE]
```

Remember to replace the `DEVICEHIGH=A:\IBMTOK.DOS` statement with the correct statement for your network adapter.

3. Create AUTOEXEC.BAT. Use the E text editor to create the following AUTOEXEC.BAT file on Diskette 1:

```

SET PATH=A:\
GOTO %CONFIG%
:NETWORK
SHARE
SET TCPHELP=A:\
SET ETCDIR=A:\
ECHO .....
ECHO Insert DLS Boot Disk 2 now, then
PAUSE
NET START
:STANDALONE

```

4. Create PROTOCOL.INI. The following sample PROTOCOL.INI file is from our IBM ValuePoint. You should copy the PROTOCOL.INI file from \NET subdirectory to Diskette 1; then use E to modify the `DHCPClientID=` parameter (highlighted below):

```

[network.setup]
version=0x3100
netcard=ibm$genibmtok,1,IBM$GENIBMTOK
transport=nts$ntst2,NTS$NTST2
transport=ibm$netbeui,IBM$NETBEUI
lana0=ibm$genibmtok,1,nts$ntst2
lana1=ibm$genibmtok,1,ibm$netbeui

[protman]
DriverName=PROTMAN$
PRIORITY=ibm$netbeui

[IBM$GENIBMTOK]
ram=0xD800
DriverName=IBMTOK$
primary

[NTS$NTST2]

```

```

DNSAddr=
GatewayAddr=
NetSubNetMask=
IPAddr=
TCPHeartBeats=Standard
DHCPClientID=BURGER
BootPFlag=DHCP
DriverName=ntsts$
VCs=16
VCReceiveLarge=6
VCSends=6
RcvWindow=2920
UseMemory=UMB
BINDINGS=IBM$GENIBMTOK
LANABASE=0
Token-Ring

```

```

[IBM$NETBEUI]
DriverName=netbeui$
SESSIONS=20
NCBS=20
BINDINGS=IBM$GENIBMTOK
LANABASE=1

```

5. Create Diskette 2. Diskette 2 contains the files required to run DLS. Label this diskette "DLS Boot Diskette 2."

Insert Diskette 2 and copy the following files from the \NET subdirectory to the diskette:

- CMDS.EXE
- CMDS16.EXE
- NET.MSG
- NET.EXE
- NETH.MSG
- NETXP.MSG
- PING.EXE
- PROTMAN.EXE

6. Create NETWORK.INI. The following sample NETWORK.INI file is from our IBM ValuePoint. You should copy the NETWORK.INI file from the \NET subdirectory to Diskette 2; then use E to modify the highlighted parameters as needed:

```

[network]
computername=BURGER
lanroot=A:\
autostart=netbeui basic

```

```

username=USERID
domain=ARMONK
autologon=no
lslogon=yes
reconnect=no
passwordcaching=no
timesync=no

[install]
peer=no
gui=no
windows=no
protocol=tcpnetr
minidls=no                **for minidls only
installed=no              **for minidls only
target=c:\WSRCLNT\        **for minidls only

[Password Lists]

```

7. Final configuration. To make Diskette 2 work correctly, you must copy the AUTOEXEC.BAT and COMMAND.COM files from Diskette 1. These files should be identical on both diskettes.

By having these two files on both diskettes, you will be able to remove Diskette 1 and replace it with Diskette 2 when the `PAUSE` statement in AUTOEXEC.BAT is executed. This allows DLS to start properly.

8. Testing. Boot the computer with Diskette 1 inserted. The following screen should be displayed:

```

PC DOS 2000 Startup Menu
=====

  1. Connect to the network
  2. No network connection

Enter a choice:          Time remaining: 15

F5=Bypass startup files F8=Confirm each line of CONFIG.SYS and AUTOEXEC.BAT [N]

```

Option 1 will be highlighted and will be automatically selected after 15 seconds. If selected, option 2 will exit directly to the DOS prompt.

When option 1 is selected, the network drivers will be loaded, and you will be prompted to insert Diskette 2 and press any key to continue.

When this is done, DLS will be started and the DOS prompt will be displayed. Note that you are not logged onto the LAN.

To check the IP configuration of the machine, use the `PING` command with the `DHCPClientID` from `NETWORK.INI` as a parameter.

```
A:\>ping burger
PING - ICMP Echo Request/Reply 2.09 (960320)
Copyright (c) 1995-1996 Network Telesystems, Inc. All rights reserved.
PING burger (192.168.6.15): 56 ICMP data bytes
64 bytes from burger: icmp_seq = 0. time < 55 ms

---- burger PING Statistics ----
1 packets transmitted
1 packets received
0% packet loss
round-trip (ms) min/avg/max = 55/55/55
```

A.2.3 Notes on the Step-by-Step Procedure

1. The test system was running IBM PC-DOS 2000.
2. The test system was running the DLS code available with the IP08267 FixPak for Warp Server, which was downloaded from <ftp://ftp.software.ibm.com/ps/products/lan/fixes/lsv5.0/english-us>

Attention

The IP08267 FixPak is provided only for licensed users of OS/2 Warp Server.

A.3 Related Publications

OS/2 Warp software distribution techniques are described in two IBM publications:

The OS/2 Warp 4 CID Rapid Deployment Tools: Migration and Installation Scenarios, SG24-2012.

The OS/2 Warp 4 CID Software Distribution Guide, SG24-2010.

Appendix B. Where Is It? Internet and IBM Intranet Web Sites

The Internet contains a wealth of information on TCP/IP, and we have included several Web addresses to help you find tools, applications, and more information.

B.1 Internet Web and FTP Sites

The DHCP Frequently Asked Questions List

<http://web.syr.edu/~jmwobus/comfaqs/dhcp.faq.html>

F/X Communications (InJoy, InJoy Connect, and Tunnel/2)

<http://www.fx.dk>

IBM DB2 Universal Database Information and Support

<http://www.software.ibm.com/data>

<http://www.software.ibm.com/data/db2/db2tech>

IBM Software Choice

<http://www.software.ibm.com/swchoice>

IBM Software Support

<http://ps.software.ibm.com>

IBM OS/2 Warp Device Driver Pak On-Line

<http://service.software.ibm.com/os2ddpak/index.htm>

IBM OS2PopS Home Page

<http://www.raleigh.ibm.com/misc/os2pops>

IBM REXX Information (including versions for Windows 95 and NT)

<http://rexx.hursley.ibm.com/rexx/rexxibm.htm>

IBM Network Computing Software Updates

<http://ps.boulder.ibm.com/pbin-usa-ps/getobj.pl?pdocs-usa/softupd.html>

Network TeleSystems (Shadow IPserver)

<http://ww.nts.com>

Internet Engineering Task Force RFCs

<http://www.isi.edu/rfc-editor>

<ftp://ds.internic.net/rfc>

IBM Hursley Labs (Java fixes)

<ftp://ftp.hursley.ibm.com>

<ftp://ftp.hursley.ibm.com/pub/java/fixes/os2/11/114>

IBM OS/2 Warp TCP/IP V4.1 Stack Updates Page

<http://service.software.ibm.com/pbin-uas-ps/getobj.pl?pdocs-usa/latest41.html>

"Hobbes" OS/2 Warp Program Library

<ftp://ftp-os2.nmsu.edu/pub/os2>

Info-ZIP's Home Page

<http://www.cdrom.com/pub/infozip>

Domino Go Webserver

<http://www.software.ibm.com/webserver/dgw/prodlist.htm>

The Fourth Crusade

<http://www.fourthcrusade.com>

Association for Computing Machinery

<http://www.acm.org>

Institute of Electrical and Electronics Engineers

<http://www.ieee.org>

LanOptics Corporation (provider of firewalls and more; OS/2 Warp version available upon request)

<http://www.lanoptics.com>

Opera Software (Web browser)

<http://www.operasoft.com>

B.2 IBM Intranet Web Sites

To reduce network traffic for everyone else, IBM employees should use the following internal Web addresses:

IBM Software Choice

<http://antero.boulder.ibm.com/asd-bin/doc>

IBM Network Computing Software Updates

<http://os2service.austin.ibm.com>

Appendix C. Application Issues

This appendix describes some specific application issues you should be aware of when using DHCP and/or DDNS. Generally speaking, most applications should not have any problem running on a DHCP client machine. There are exceptions, however. We recommend that you consult with software vendors to make sure that your TCP/IP applications work well on a DHCP client system. If necessary, you can configure your client machine with a static IP address if you must run a particular application.

Many applications (such as network management utilities) can be much easier to use with a Dynamic DNS server on the network if you do rely on DHCP. Otherwise, users may find it difficult to locate other systems on the network.

We have tested several popular applications with DHCP and DDNS technologies, and, while the overwhelming majority of applications have no problems, we have found some minor difficulties with a few specific applications. The following sections describe those problems.

C.1 DB2 Universal Database

Many network managers prefer to assign fixed IP addresses to major servers to prevent them from relying on other servers in order to function properly. Quite often, DB2 servers have fixed IP addresses.

However, the client version of DB2 may be installed on a workstation which receives its address assignment from a DHCP server. We found a problem with DB2 UDB Version 5 on OS/2 Warp DHCP clients. (This problem will be fixed in the next version of DB2. To keep abreast of DB2 technical information, including any resolution to this problem, you should visit:

<http://www.software.ibm.com/data/db2/db2tech>

on the Internet.)

To identify this problem, we installed IBM's DB2 Universal Database Personal Edition on an OS/2 Warp client. The INSTALL.TXT file warns that the TCP/IP protocol stack will not be recognized on a DHCP client. In fact, DB2 itself seems to work well on such a client. We could only find one problem: The HTML online documentation system (NET.QUESTION) did not work properly on a DHCP client.

The Windows 95 and Windows NT versions of DB2 UDB also contain the same warning in INSTALL.TXT. However, we experienced no obvious problems when installing or running DB2 UDB on a Windows 95 or Windows NT DHCP client.

C.2 TME 10 Framework

TME 10 needs to have the correct IP addresses for managed nodes and endpoints in order to work properly. Currently, TME 10 has two ways to provide client support for DHCP: the UserLink/DHCP Service component of PC Agent and LCF (Lightweight Client Framework). (If possible, you should use Tivoli's LCF since it provides newer technology and will gradually replace the UserLink/DHCP Service. LCF, which many refer to as a "thin client" product, takes up very little disk space.)

Generally, the Tivoli service, a daemon, runs on each managed client and server attached to the network. As these daemons try to connect with one another, they must be able to map a system's IP address to a valid hostname. This reverse mapping technique may not be available in a DHCP-only environment, and many ordinary DNS servers can run without reverse mapping (for instance, no PTR record update). In fact, Tivoli products use both hostname to IP address mapping and reverse mapping, and installation of TME 10 clients will fail if both services are not provided on your network.

Historically, Tivoli has recommended a Microsoft Windows NT WINS server to help map IP addresses to client computer names (at least for WINS clients such as Windows 95 and Windows NT, where the host and computer names are often identical). However, a DDNS server can provide broader support for DHCP LCF clients.

For more information on DDNS and DHCP issues with TME 10, please refer to the following publications:

TME 10 Framework Version 3.2: An Introduction to the Lightweight Client Framework, SG24-2025.

TME 10 Internals and Problem Determination, SG24-2034.

C.3 NetFinity

NetFinity Version 5 (and later), IBM's PC management software for workgroups, works well with DHCP clients. You also need a DNS (preferably

DDNS) server, and you need to keep in mind some minor configuration issues. The following sections describe those issues.

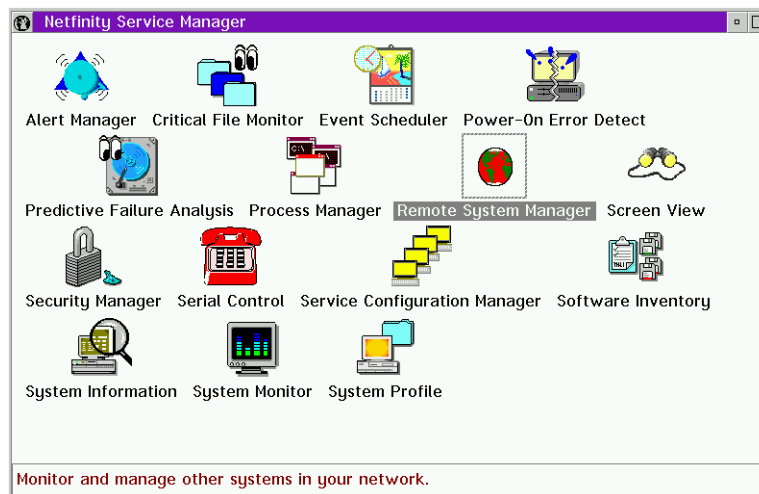


Figure 253. [NetFinity] NetFinity Service Manager Folder

C.3.1 Manager

NetFinity Manager has to be explicitly configured to use dynamic addressing. First, open the **NetFinity Service Manager**. Then open the **Remote System Manager** as shown in Figure 253. The System Group Management folder should appear with a single default group, All. Click on **Options** and then on **Dynamic Address Options** as shown in Figure 254. When the Dynamic Address Options window opens, check the checkboxes as shown in Figure 255. (You may also wish to increase the Dynamic Ping Interval from the default of 1 minute to 3 minutes.) Click **OK** to save, and you are finished.

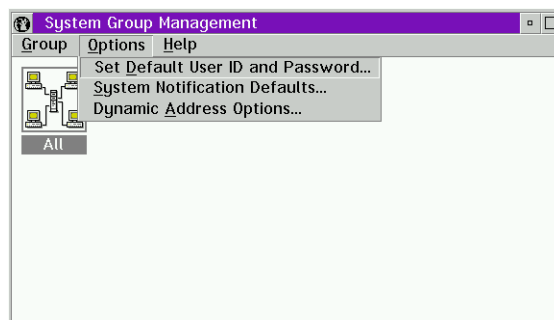


Figure 254. [NetFinity] System Group Management Options

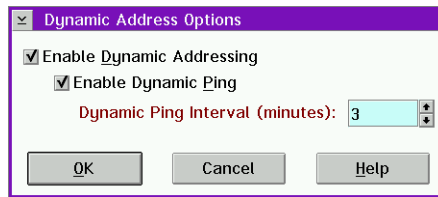


Figure 255. [NetFinity] Dynamic Address Options

C.3.2 Client

The client configuration simply requires coordination between the NetFinity system name (for TCP/IP) and the designated IP hostname for the system. As long as these two names are the same, NetFinity Manager can track the machine by name, even if its IP address changes.

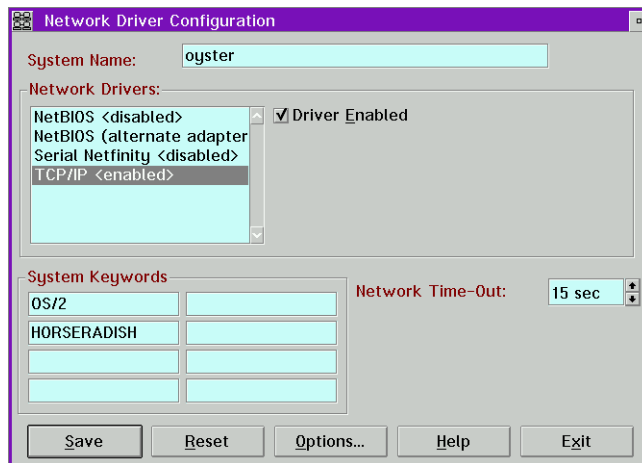


Figure 256. [NetFinity] Network Driver Configuration for a Client

To configure the NetFinity client, first open the NetFinity folder and then open the **Network Driver Configuration**. The Network Driver Configuration window (Figure 256) allows you to change certain client settings. You should make sure that the System Name is set the same as the IP hostname for the client (as registered with the name server). Also, the TCP/IP driver should be enabled.

You may wish to add some System Keywords to more easily identify client systems on the network. You can assign keywords for different categories of clients, by client operating system, processor type, location, department, and so forth.

Click on **Save** and you should see a window similar to Figure 257. Click on **OK** then **Exit**. Shut

down and reboot the system so that the changes you made can take effect, or simply kill and restart NETFBASE.EXE.

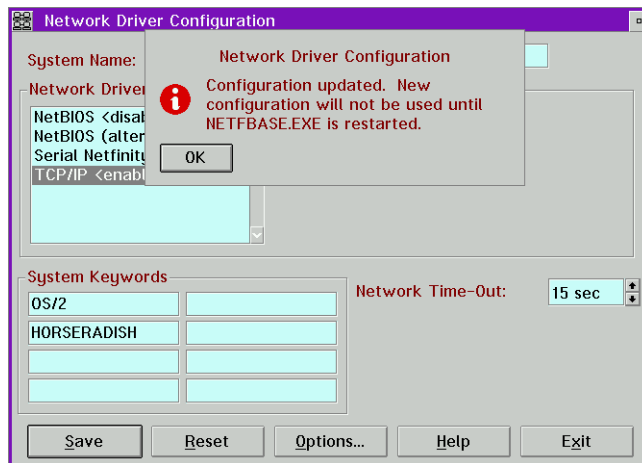


Figure 257. [NetFinity] Network Driver Configuration Updated

You should also configure the DDNS client hostname to be identical to the NetFinity name. How you register your client with the DDNS server will depend on the type of client. For example, in OS/2 Warp 4, you can use the DDNS Client Configuration program (Figure 258). Note that the hostname "oyster" is identical to the NetFinity client name.

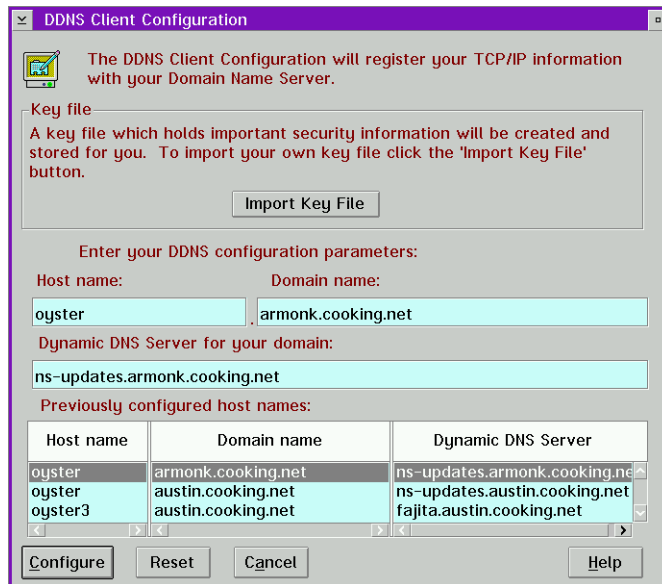


Figure 258. [OS/2 Warp] DDNS Client Configuration

Once you have completed setup of your NetFinity client(s), you should be able to monitor them using the NetFinity Manager.

C.4 Netscape Navigator

Netscape's popular Web browser has no trouble working on any client receiving its address assignment from a DHCP server. In fact, some versions of Netscape Navigator can be enhanced using DHCP technologies. Please see 7.2.3, "Netscape "Message of the Day" Service" on page 295, and 7.2.4, "Custom Netscape INI Files" on page 302, for more information on Netscape Navigator and DHCP.

Appendix D. CD-ROM Contents

We have included additional utilities and information on the CD-ROM accompanying this book. You can also avoid typing in long script files or other examples, since these too are included on the CD-ROM.

You can browse the contents of the CD-ROM by opening the \INDEX.HTM file with your favorite Web browser, such as Netscape:

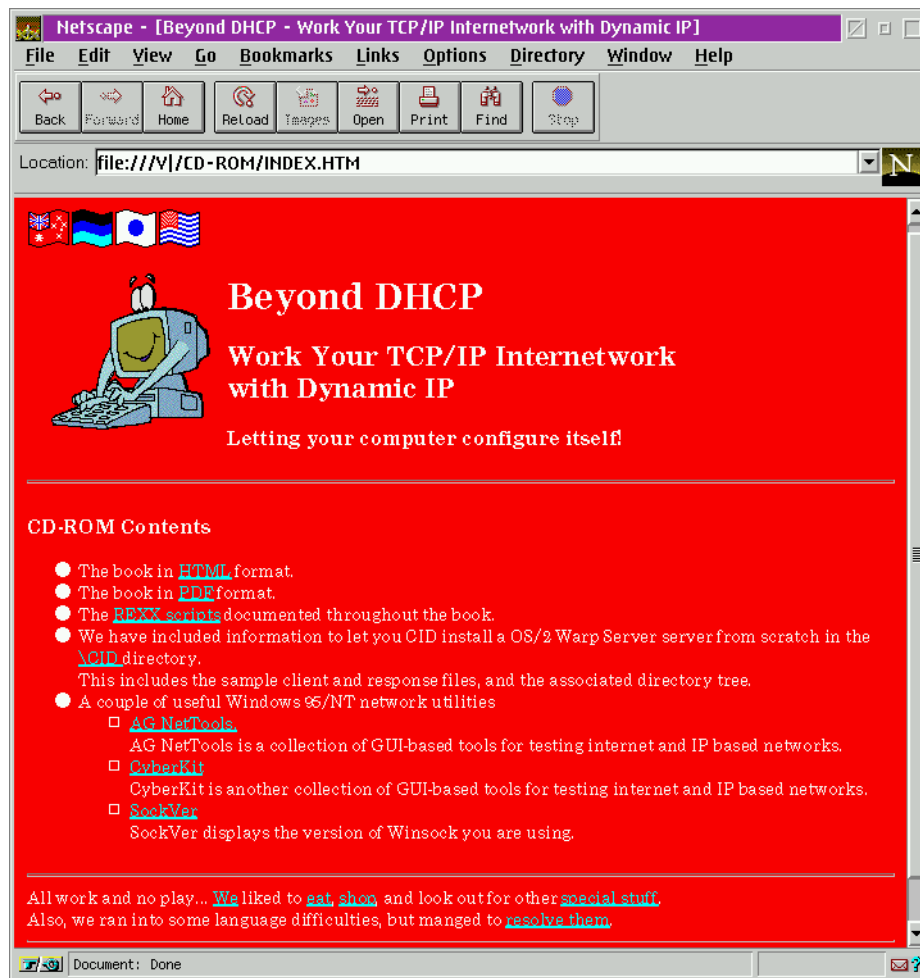


Figure 259. CD-ROM Contents: INDEX.HTM

Note: The shown URL of file:///V:/CD-ROM/INDEX.HTM represents a networked drive and directory.

Here's what's on the CD-ROM

- The book in HTML format.
- The book in Acrobat Reader PDF format.
- The REXX scripts documented throughout the book.
- Information to let you CID install a OS/2 Warp Server server from scratch can be found in the \CID directory. Included is:
 - A Client .CMD file to install a Warp Server server from scratch.
 - Response files for:
 - Java 1.1.4
 - OS/2 Warp Server File and Print Services (LAN Server 5)
 - MPTS 5.30
 - Netscape 2.02
 - OS/2 Version 3.x
 - TCP/IP Version 4.1 for OS/2
 - IP08506 CSD OS/2 Warp Server File and Print Services
 - FixPak 35 for OS/2 3.x
 - Fixes and CSD's included:
 - Fixes for JAVA 1.1.4
 - Fixes for MPTS 5.30
 - Fixes for TCP/IP 4.1
 - IP08506 CSD for OS/2 Warp Server File and Print Services (LAN Server 5)
 - IP_8508 CSD for HPFS386
 - FixPak 35 for OS/2 Version 3.x
- A couple of useful Windows 95/NT network utilities
 - AG NetTools -A collection of GUI-based tools for testing Internet- and IP-based networks.
 - CyberKit - Another collection of GUI-based tools for testing Internet- and IP-based networks.
 - SockVer - Displays the version of Winsock you are using.

Appendix E. DHCP Options (RFC 2132)

The Internet Engineering Task Force (IETF) publishes the official documents describing Internet standards, including DHCP. DHCP provides several different options. This appendix, based on RFC 2132, includes a complete list of those options, along with information on which options are supported by popular server and client operating systems.

DHCP provides a framework for passing configuration information to hosts on a TCP/IP network. Configuration parameters and other control information are carried in tagged data items that are stored in the "options" field of the DHCP message. The data items themselves are also called "options."

RFC 1497 was published previously and included vendor information extensions. These extensions are also described in this appendix, and RFC 1497 should now be considered obsolete. In addition, all the DHCP options described here, except those specific to DHCP in Part E.3, "DHCP (Only) Options" on page 432, may be used as BOOTP vendor information extensions.

For the latest, official list of DHCP options please visit:

`ftp://ds.internic.net/rfc/rfc2132.txt`

on the Internet.

E.1 Introduction

DHCP options have the same format as the BOOTP "vendor extensions" defined in RFC 1497. Options may be fixed length or variable length. All options begin with a tag octet, which uniquely identifies the option. Fixed length options without data consist of only a tag octet. Only options 0 and 255 are fixed length. All other options are variable length with a length octet following the tag octet. The value of the length octet does not include the two octets specifying the tag and length. The length octet is followed by (length) octets of data. With some variable length options the length field is a constant but must still be specified. Any options defined in a future version of this standard must contain a length octet even if the length is fixed or zero.

Options containing NVT ASCII data should not include a trailing null. However, the receiver of such options must be prepared to delete trailing nulls if they exist. The receiver must not require that a trailing null be included in the data.

Option codes 128 to 254 (decimal) are reserved for your own use.

E.2 DHCP and BOOTP Options

This section introduces all DHCP and BootP options that can be defined at the DHCP server and delivered to requesting DHCP and BootP clients. Be aware that although most DHCP servers can deliver all options illustrated here to requesting DHCP and BootP clients, there are platform-specific dependencies that determine whether or not requesting DHCP and BootP clients can actually make use of all delivered information.

E.2.1 Options 0 and 255: Pad and End

Originally described in RFC 1497, options 0 and 255 help provide vendor extensions to the DHCP standard.

Option 0, the Pad Option, can be used to cause subsequent fields to align on word boundaries. Its length is one octet. The End Option, option 255, marks the end of valid information in the vendor field. (Subsequent octets should be filled with pad options.) Its length is also one octet.

E.2.2 Option 1: Subnet Mask

Option 1 specifies the client's subnet mask per RFC 950. If both the subnet mask and the router (option 3) are specified in a DHCP reply, the subnet mask option must be first.

The code for the subnet mask option is 1, and its length is 4 octets.

Code	Length	Subnet Mask			
1	4	m1	m2	m3	m4

E.2.3 Option 2: Time Offset

The time offset field specifies the offset, in seconds, of the client's subnet from Coordinated Universal Time (UTC). The offset is expressed as a two's complement 32-bit integer. A positive offset indicates a location east of the zero meridian, and a negative offset indicates a location west of the zero meridian.

The code for the time offset option is 2, and its length is 4 octets.

Code	Length	Time Offset			
1	4	n1	n2	n3	n4

E.2.4 Option 3: Router

The router option specifies a list of IP addresses for routers on the client's subnet. Routers should be listed in order of preference.

The code for the router option is 3, and the minimum length for the router option is 4 octets. The length must always be a multiple of 4.

Code	Length	Address 1				Address 2 [...]			
3	n	a1	a2	a3	a4	a1	a2	[...]	

E.2.5 Option 4: Time Server

The time server option specifies a list of RFC 868 time servers available to the client. Servers should be listed in order of preference.

The code for the time server option is 4, and the minimum length for this option is 4 octets. The length must always be a multiple of 4.

Code	Length	Address 1				Address 2 [...]			
4	n	a1	a2	a3	a4	a1	a2	[...]	

E.2.6 Option 5: IEN 116 (Old) Name Server

The name server option specifies a list of IEN 116 name servers available to the client. Servers should be listed in order of preference. (Option 5 is provided for compatibility with old-style name servers. Use Option 6 for modern domain name servers.)

The code for the name server option is 5, and the minimum length for this option is 4 octets. The length must always be a multiple of 4.

Code	Length	Address 1				Address 2 [...]			
5	n	a1	a2	a3	a4	a1	a2	[...]	

E.2.7 Option 6: Domain Name Server

The domain name server option specifies a list of Domain Name System (RFC 1035) name servers available to the client. Servers should be listed in order of preference.

The code for the domain name server option is 6, and the minimum length for this option is 4 octets. The length must always be a multiple of 4.

Code	Length	Address 1				Address 2				[...]
6	n	a1	a2	a3	a4	a1	a2	a3	a4	[...]

E.2.8 Option 7: Log Server

The log server option specifies a list of MIT-LCS UDP log servers available to the client. Servers should be listed in order of preference.

The code for the log server option is 7, and the minimum length for this option is 4 octets. The length must always be a multiple of 4.

Code	Length	Address 1				Address 2				[...]
7	n	a1	a2	a3	a4	a1	a2	a3	a4	[...]

E.2.9 Option 8: Cookie Server

The cookie server option specifies a list of RFC 865 cookie servers available to the client. Servers should be listed in order of preference.

The code for the log server option is 8, and the minimum length for this option is 4 octets. The length must always be a multiple of 4.

Code	Length	Address 1				Address 2				[...]
8	n	a1	a2	a3	a4	a1	a2	a3	a4	[...]

E.2.10 Option 9: LPR Server

The LPR server option specifies a list of RFC 1179 print servers available to the client. Servers should be listed in order of preference.

The code for the LPR server option is 9, and the minimum length for this option is 4 octets. The length must always be a multiple of 4.

Code	Length	Address 1				Address 2 [...]			
9	n	a1	a2	a3	a4	a1	a2	[...]	

E.2.11 Option 10: Impress Server

Option 10 specifies a list of Imagen Impress servers available to the client. Servers should be listed in order of preference.

The code for the Impress server option is 10, and the minimum length for this option is 4 octets. The length must always be a multiple of 4.

Code	Length	Address 1				Address 2 [...]			
10	n	a1	a2	a3	a4	a1	a2	[...]	

E.2.12 Option 11: Resource Location Server

This option specifies a list of RFC 887 resource location servers available to the client. Servers should be listed in order of preference.

The code for this option is 11, and the minimum length for this option is 4 octets. The length must always be a multiple of 4.

Code	Length	Address 1				Address 2 [...]			
11	n	a1	a2	a3	a4	a1	a2	[...]	

E.2.13 Option 12: Host Name

This option specifies the name of the client. The name may or may not include the full local domain name. (See E.2.16, “Option 15: Domain Name” on page 418, for the preferred method of obtaining the domain name.)

RFC 1035 explains the restrictions concerning which characters can be used in the host name.

The code for this option is 12, and its minimum length is 1.

Code	Length	Host Name							
12	n	h1	h2	h3	h4	h5	h6	[...]	

E.2.14 Option 13: Boot File Size

This option specifies the length, in 512-octet blocks, of the default boot image file for the client. The file length is specified as an unsigned 16-bit integer.

The code for this option is 13, and its length is 2.

Code	Length	File Size
13	2	11 12

E.2.15 Option 14: Merit Dump File Name

This option specifies the file name (with path) of a file where the client's core image should be dumped in the event the client crashes. The path is formatted as a string consisting of characters from the NVT ASCII character set.

The code for this option is 14, and its minimum length is 1.

Code	Length	Dump Path/File Name
14	n	n1 n2 n3 n4 [...]

E.2.16 Option 15: Domain Name

This option specifies the default domain name that the client should use when resolving host names using the Domain Name System.

The code for this option is 15, and its minimum length is 1.

Code	Length	Domain Name
15	n	d1 d2 d3 d4 [...]

E.2.17 Option 16: Swap Server

This specifies the IP address of the client's swap server.

The code for this option is 16, and its length is 4.

Code	Length	Swap Server Address
16	n	a1 a2 a3 a4

E.2.18 Option 17: Root Path

This option specifies the path name that contains the client's root directory. The path is formatted as a string consisting of characters from the NVT ASCII character set.

The code for this option is 17, and its minimum length is 1.

Code	Length	Root Disk Pathname				
17	n	n1	n2	n3	n4	[...]

E.2.19 Option 18: Extensions Path

This option specifies the path and name of a file, retrievable through TFTP, which contains information to be interpreted in the same way as the 64-octet vendor extension field within the BOOTP response. The following exceptions apply:

- The length of the file is not limited.
- All references to Tag 18 (instances of the BOOTP extensions path field) within the file are ignored.

The code for this option is 18, and its minimum length is 1.

Code	Length	Extensions Path/File Name				
18	n	n1	n2	n3	n4	[...]

E.2.20 Option 19: IP Forwarding Enable/Disable

This option specifies whether the client should configure its IP layer for packet forwarding. A value of 0 means disable IP forwarding, and a value of 1 means enable IP forwarding.

The code for this option is 19, and its length is 1.

Code	Length	Value
19	1	n

E.2.21 Option 20: Non-Local Source Routing Enable/Disable

This option specifies whether the client should configure its IP layer to allow forwarding of datagrams with non-local source routes. A value of 0 means

disallow forwarding of such datagrams, and a value of 1 means allow forwarding.

The code for this option is 20, and its length is 1.

Code	Length	Value
20	1	n

E.2.22 Option 21: Policy Filter

This option specifies policy filters for non-local source routing. The filters consist of a list of IP addresses and masks which specify destination/mask pairs with which to filter incoming source routes.

Any source routed datagram whose next hop address does not match one of the filters should be discarded by the client.

The code for this option is 21, and the minimum length of this option is 8. The length must be a multiple of 8.

Code	Length	Address 1				Mask 1			
21	n	a1	a2	a3	a4	m1	m2	m3	m4

Address 2				Mask 2			
a1	a2	a3	a4	m1	m2	m3	m4

E.2.23 Option 22: Maximum Datagram Reassembly Size

This option specifies the maximum size datagram that the client should be prepared to reassemble. The size is specified as a 16-bit unsigned integer. The minimum legal value is 576.

The code for this option is 22, and its length is 2.

Code	Length	Size	
22	2	s1	s2

E.2.24 Option 23: Default IP Time-to-Live

This option specifies the default time-to-live that the client should use for outgoing datagrams. The TTL is specified as an octet with a value between 1 and 255.

The code for this option is 23, and its length is 1.

Code	Length	TTL
23	1	ttl

E.2.25 Option 24: Path MTU Aging Timeout

This option specifies the timeout (in seconds) to use when aging path MTU values discovered by the mechanism defined in RFC 1191. The timeout is specified as a 32-bit unsigned integer.

The code for this option is 24, and its length is 4.

Code	Length	Timeout
24	4	t1 t2 t3 t4

E.2.26 Option 25: Path MTU Plateau Table

This option specifies a table of MTU sizes to use when performing path MTU discovery as defined in RFC 1191. The table is formatted as a list of 16-bit unsigned integers, ordered from smallest to largest. The minimum MTU value cannot be smaller than 68.

The code for this option is 25, and its minimum length is 2. The length must be a multiple of 2.

Code	Length	Size 1	Size 2
25	n	s1 s2	s1 s2 [...]

E.2.27 Option 26: Interface MTU

This option specifies the MTU to use for this interface. The MTU is specified as a 16-bit unsigned integer. The minimum legal value for the MTU is 68.

The code for this option is 26, and its length is 2.

Code	Length	MTU
26	2	m1 m2

E.2.28 Option 27: All Subnets Are Local

This option specifies whether or not the client may assume that all subnets of the IP network to which the client is connected use the same MTU as the subnet of that network to which the client is directly connected. A value of 1 indicates that all subnets share the same MTU. A value of 0 means that the client should assume that some subnets of the directly connected network may have smaller MTUs.

The code for this option is 27, and its length is 1.

Code	Length	Value
27	1	n

E.2.29 Option 28: Broadcast Address

This option specifies the broadcast address in use on the client's subnet.

The code for this option is 28, and its length is 4.

Code	Length	Broadcast Address
28	4	b1 b2 b3 b4

E.2.30 Option 29: Perform Mask Discovery

This option specifies whether or not the client should perform subnet mask discovery using ICMP. A value of 0 indicates that the client should not perform mask discovery. A value of 1 means that the client should perform mask discovery.

The code for this option is 29, and its length is 1.

Code	Length	Value
29	1	n

E.2.31 Option 30: Mask Supplier

This option specifies whether or not the client should respond to subnet mask requests using ICMP. A value of 0 indicates that the client should not respond. A value of 1 means that the client should respond.

The code for this option is 30, and its length is 1.

Code	Length	Value
30	1	n

E.2.32 Option 31: Perform Router Discovery

This option specifies whether or not the client should solicit routers using the router discovery mechanism defined in RFC 1256. A value of 0 indicates that the client should not perform router discovery. A value of 1 means that the client should perform router discovery.

The code for this option is 31, and its length is 1.

Code	Length	Value
31	1	n

E.2.33 Option 32: Router Solicitation Address

This option specifies the address to which the client should transmit router solicitation requests.

The code for this option is 32, and its length is 4.

Code	Length	Address
32	4	a1 a2 a3 a4

E.2.34 Option 33: Static Route

This option specifies a list of static routes that the client should install in its routing cache. If multiple routes to the same destination are specified, they are listed in descending order of priority.

The routes consist of a list of IP address pairs. The first address is the destination address, and the second address is the router for the destination. The default route (0.0.0.0) is an illegal destination for a static route.

The code for this option is 33, and the minimum length for this option is 8. The length must be a multiple of 8.

Code		Length		Destination 1				Router 1			
+	+	+	+	+	+	+	+	+	+	+	+
	33		n		d1		d2		d3		d4
	r1		r2		r3		r4				
+	+	+	+	+	+	+	+	+	+	+	+

Destination 2				Router 2							
+	+	+	+	+	+	+	+	+	+	+	
	d1		d2		d3		d4		r1		r2
									r3		r4
									[...]		
+	+	+	+	+	+	+	+	+	+	+	+

E.2.35 Option 34: Trailer Encapsulation

This option specifies whether or not the client should negotiate the use of trailers (RFC 893) when using the ARP protocol. A value of 0 indicates that the client should not attempt to use trailers. A value of 1 means that the client should attempt to use trailers.

The code for this option is 34, and its length is 1.

Code	Length	Value
34	1	n

E.2.36 Option 35: ARP Cache Timeout

This option specifies the timeout, in seconds, for ARP cache entries. The time is specified as a 32-bit unsigned integer.

The code for this option is 35, and its length is 4.

Code	Length	Time			
35	4	t1	t2	t3	t4

E.2.37 Option 36: Ethernet Encapsulation

This option specifies whether or not the client should use Ethernet Version 2 (RFC 894) or IEEE 802.3 (RFC 1042) encapsulation if the interface is Ethernet. A value of 0 indicates that the client should use RFC 894 encapsulation. A value of 1 means that the client should use RFC 1042 encapsulation.

The code for this option is 36, and its length is 1.

Code	Length	Value
36	1	n

E.2.38 Option 37: TCP Default Time-to-Live

This option specifies the default time-to-live that the client should use when sending TCP segments. The value is represented as an 8-bit unsigned integer. The minimum value is 1.

The code for this option is 37, and its length is 1.

Code	Length	TTL
37	1	n

E.2.39 Option 38: TCP Keepalive Interval

This option specifies the interval, in seconds, that the client should wait before sending a keepalive message on a TCP connection. The time is specified as a 32-bit unsigned integer. A value of zero indicates that the client should not generate keep-alive messages on connections unless specifically requested by an application.

The code for this option is 38, and its length is 4.

Code	Length	Time
38	4	t1 t2 t3 t4

E.2.40 Option 39: TCP Keepalive Garbage

This option specifies whether or not the client should send TCP keep-alive messages with an octet of garbage for compatibility with older implementations. A value of 0 indicates that a garbage octet should not be sent. A value of 1 indicates that a garbage octet should be sent.

The code for this option is 39, and its length is 1.

Code	Length	Value
39	1	n

E.2.41 Option 40: Network Information Service Domain

This option specifies the name of the client's NIS domain. The domain is formatted as a string consisting of characters from the NVT ASCII character set.

The code for this option is 40, and its minimum length is 1.

Code	Length	NIS Domain Name				
40	n	n1	n2	n3	n4	[...]

E.2.42 Option 41: Network Information Server

This option specifies a list of IP addresses indicating NIS servers available to the client. Servers should be listed in order of preference.

The code for this option is 41, and its minimum length is 4. The length must be a multiple of 4.

Code	Length	Address 1				Address 2			
41	n	a1	a2	a3	a4	a1	a2	[...]	

E.2.43 Option 42: Network Time Protocol Server

This option specifies a list of IP addresses indicating NTP servers available to the client. Servers should be listed in order of preference.

The code for this option is 42, and its minimum length is 4. The length must be a multiple of 4.

Code	Length	Address 1				Address 2			
42	n	a1	a2	a3	a4	a1	a2	[...]	

E.2.44 Option 43: Vendor-Specific Information

This option is used by clients and servers to exchange vendor-specific information. The information is an opaque object of *n* octets, presumably interpreted by vendor-specific code on the clients and servers. The definition of this information is vendor-specific.

The vendor is indicated in the vendor class identifier option. Servers not equipped to interpret the vendor-specific information sent by a client must ignore it, although it may be reported.

Clients which do not receive desired vendor-specific information should make an attempt to operate without it, although they may do so, and announce they are doing so, with degraded functionality.

If a vendor potentially encodes more than one item of information in this option, then the vendor should encode the option using "encapsulated vendor-specific options." The encapsulated vendor-specific options field should be encoded as a sequence of code/length/value fields of identical syntax to the DHCP options field with the following exceptions:

- There should not be a "magic cookie" field in the encapsulated vendor-specific extensions field.
- Codes other than 0 or 255 may be redefined by the vendor within the encapsulated vendor-specific extensions field, but should conform to the tag/length/value syntax.
- Code 255 (End Option), if present, signifies the end of the encapsulated vendor extensions, not the end of the vendor extensions field. If no code 255 is present, then the end of the enclosing vendor-specific information field is taken as the end of the encapsulated vendor-specific extensions field.

The code for this option is 43, and its minimum length is 1.

Code	Length	Vendor-Specific Information		
43	n	i1	i2	[...]

When encapsulated vendor-specific extensions are used, the information bytes (i_1 to i_n) have the following format:

Code	Length	Data	Item	Code	Length	Data	Item	Code
T1	n	d1	d2	...	T2	n	D1	D2
			

E.2.45 Option 44: NetBIOS over TCP/IP Name Server Option

The NetBIOS name server (NBNS) option specifies a list of RFC 1001/1002 NBNS name servers, such as Shadow IPserver, listed in order of preference.

The code for this option is 44, and the minimum length of the option is 4 octets. The length must always be a multiple of 4.

Code	Length	Address 1				Address 2			
44	n	a1	a2	a3	a4	b1	b2	b3	b4 [...]

E.2.46 Option 45: NetBIOS over TCP/IP Datagram Distribution Server

The NetBIOS datagram distribution server (NBDD) option specifies a list of RFC 1001/1002 NBDD servers listed in order of preference.

The code for this option is 45, and the minimum length of the option is 4 octets. The length must always be a multiple of 4.

Code	Length	Address 1				Address 2			
45	n	a1	a2	a3	a4	b1	b2	b3	b4 [...]

E.2.47 Option 46: NetBIOS over TCP/IP Node Type

The NetBIOS node type option allows NetBIOS over TCP/IP clients to be configured as described in RFC 1001/1002. The value is specified as a single octet which identifies the client type as follows:

Value (Hex)	Node Type
01	B-node
02	P-node
04	M-node
08	H-node

The code for this option is 46, and the length of this option is always 1.

Code	Length	Node Type
46	1	See Above

E.2.48 Option 47: NetBIOS over TCP/IP Scope

This option specifies the NetBIOS over TCP/IP scope parameter for the client as described in RFC 1001/1002. (Certain restrictions on the characters that can be used may apply.)

The code for this option is 47, and the minimum length of this option is 1.

Code	Length	NetBIOS Scope			
47	n	s1	s2	s3	s4 [...]

E.2.49 Option 48: X Window System Font Server Option

This option specifies a list of X-Window System font servers available to the client. Servers should be listed in order of preference.

The code for this option is 48, and the minimum length of this option is 4 octets. The length must be a multiple of 4.

Code	Length	Address 1				Address 2			
48	n	a1	a2	a3	a4	a1	a2	a3	a4 [...]

E.2.50 Option 49: X Window System Display Manager

This option specifies a list of IP addresses of systems that are running the X Window System Display Manager and are available to the client. Addresses should be listed in order of preference.

The code for this option is 49, and the minimum length of this option is 4. The length must be a multiple of 4.

Code	Length	Address 1				Address 2			
49	n	a1	a2	a3	a4	a1	a2	a3	a4 [...]

E.2.51 Option 64: Network Information Service (Plus) Domain

This option specifies the name of the client's NIS+ domain. The domain is formatted as a string consisting of characters from the NVT ASCII character set.

The code for this option is 64, and its minimum length is 1.

Code	Length	NIS+ Domain Name			
64	n	n1	n2	n3	n4 [...]

E.2.52 Option 65: Network Information Service (Plus) Server

This option specifies a list of IP addresses indicating NIS+ servers available to the client. Servers should be listed in order of preference.

The code for this option is 65, and its minimum length is 4. The length must be a multiple of 4.

Code	Length	Address 1				Address 2			
65	n	a1	a2	a3	a4	a1	a2	[...]	

E.2.53 Option 68: Mobile IP Home Agent

This option specifies a list of IP addresses indicating mobile IP home agents available to the client. Agents should be listed in order of preference.

The code for this option is 68. Its minimum length is 0, indicating no home agents are available. The length must be a multiple of 4. The usual length will be 4 octets, containing a single home agent's address.

Code	Length	Address 1				
68	n	a1	a2	a3	a4	[...]

E.2.54 Option 69: Simple Mail Transport Protocol (SMTP) Server

This option specifies a list of SMTP servers available to the client. Servers should be listed in order of preference.

The code for the SMTP server option is 69, and the minimum length for this option is 4 octets. The length must always be a multiple of 4.

Code	Length	Address 1				Address 2			
69	n	a1	a2	a3	a4	a1	a2	[...]	

E.2.55 Option 70: Post Office Protocol (POP3) Server

This option specifies a list of POP3 mail servers available to the client. Servers should be listed in order of preference.

The code for the POP3 server option is 70, and the minimum length for this option is 4 octets. The length must always be a multiple of 4.

Code	Length	Address 1				Address 2			
70	n	a1	a2	a3	a4	a1	a2	[...]	

E.2.56 Option 71: Network News Transport Protocol (NNTP) Server

This option specifies a list of NNTP servers available to the client. Servers should be listed in order of preference.

The code for the NNTP server option is 71, and the minimum length for this option is 4 octets. The length must always be a multiple of 4.

Code	Length	Address 1				Address 2			
71	n	a1	a2	a3	a4	a1	a2	[...]	

E.2.57 Option 72: Default World Wide Web (WWW) Server

This option specifies a list of WWW servers available to the client. Servers should be listed in order of preference.

The code for the WWW server option is 72, and the minimum length for this option is 4 octets. The length must always be a multiple of 4.

Code	Length	Address 1				Address 2			
72	n	a1	a2	a3	a4	a1	a2	[...]	

E.2.58 Option 73: Default Finger Server

This option specifies a list of Finger servers available to the client. Servers should be listed in order of preference.

The code for the Finger server option is 73, and the minimum length for this option is 4 octets. The length must always be a multiple of 4.

Code	Length	Address 1				Address 2			
73	n	a1	a2	a3	a4	a1	a2	[...]	

E.2.59 Option 74: Default Internet Relay Chat (IRC) Server

This option specifies a list of IRC servers available to the client. Servers should be listed in order of preference.

The code for the IRC server option is 74, and the minimum length for this option is 4 octets. The length must always be a multiple of 4.

Code	Length	Address 1				Address 2			
74	n	a1	a2	a3	a4	a1	a2	[...]	

E.2.60 Option 75: StreetTalk Server

This option specifies a list of StreetTalk servers available to the client. Servers should be listed in order of preference.

The code for the StreetTalk server option is 75, and the minimum length for this option is 4 octets. The length must always be a multiple of 4.

Code	Length	Address 1				Address 2			
75	n	a1	a2	a3	a4	a1	a2	[...]	

E.2.61 Option 76: StreetTalk Directory Assistance (STDA) Server

This option specifies a list of StreetTalk Directory Assistance servers available to the client. Servers should be listed in order of preference.

The code for the StreetTalk Directory Assistance server option is 76, and the minimum length for this option is 4 octets. The length must always be a multiple of 4.

Code	Length	Address 1				Address 2			
76	n	a1	a2	a3	a4	a1	a2	[...]	

E.3 DHCP (Only) Options

This section introduces all DHCP-only options that can be defined at the DHCP server and delivered to requesting DHCP clients. Be aware that although most DHCP servers can deliver all options illustrated here to requesting DHCP clients, there are platform-specific dependencies that determine whether or not requesting DHCP clients can actually make use of all delivered information.

E.3.1 Option 50: Requested IP Address

This option is used in a client request (DHCPDISCOVER) to allow the client to request that a particular IP address be assigned.

The code for this option is 50, and its length is 4.

Code	Length	Address			
50	4	a1	a2	a3	a4

E.3.2 Option 51: IP Address Lease Time

This option is used in a client request (DHCPDISCOVER or DHCPREQUEST) to allow the client to request a lease time for the IP address. In a server reply (DHCPOFFER), a DHCP server uses this option to specify the lease time it is willing to offer. The time is expressed in seconds and is specified as a 32-bit unsigned integer.

The code for this option is 51, and its length is 4.

Code	Length	Lease Time			
51	4	t1	t2	t3	t4

E.3.3 Option 52: Option Overload

This option is used to indicate that the DHCP "sname" or "file" fields are being overloaded by using them to carry DHCP options. A DHCP server inserts this option if the returned parameters will exceed the usual space allotted for options. If this option is present, the client interprets the specified additional fields after it finishes interpretation of the standard option fields.

The code for this option is 52, and its length is 1. Legal values for this option are:

Value	Meaning
1	the "file" field is used to hold options
2	the "sname" field is used to hold options
3	both fields are used to hold options

Code	Length	Value
52	1	n

E.3.4 Option 53: DHCP Message Type

This option is used to convey the type of the DHCP message.

The code for this option is 53, and its length is 1. Legal values for this option are:

Value	Message Type
-----	-----
1	DHCPDISCOVER
2	DHCPOFFER
3	DHCPREQUEST
4	DHCPDECLINE
5	DHCPACK
6	DHCPNAK
7	DHCPRELEASE
8	DHCPINFORM

Code	Length	Type
-----+-----+-----+		
53	1	n
-----+-----+-----+		

E.3.5 Option 54: Server Identifier

This option is used in DHCPOFFER and DHCPREQUEST messages and may optionally be included in the DHCPACK and DHCPNAK messages. DHCP servers include this option in the DHCPOFFER in order to allow the client to distinguish between lease offers. DHCP clients use the contents of the "server identifier" field as the destination address for any DHCP messages unicast to the DHCP server. DHCP clients also indicate which of several lease offers is being accepted by including this option in a DHCPREQUEST message. The identifier is the IP address of the selected server.

The code for this option is 54, and its length is 4.

Code	Length	Address
-----+-----+-----+-----+		
54	4	a1 a2 a3 a4
-----+-----+-----+-----+		

E.3.6 Option 55: Parameter Request List

This option is used by a DHCP client to request values for specified configuration parameters. The list of requested parameters is specified as (n) octets, where each octet is a valid DHCP option code as defined in this appendix.

The client may list the options in order of preference. The DHCP server is not required to return the options in the requested order but must try to insert the requested options in the order requested by the client.

The code for this option is 55, and its minimum length is 1.

Code	Length	Option Codes
55	n	c1 c2 [...]

E.3.7 Option 56: Message

This option is used by a DHCP server to provide an error message to a DHCP client in a DHCPNAK message in the event of a failure. A client may use this option in a DHCPDECLINE message to indicate why the client declined the offered parameters. The message consists of (n) octets of NVT ASCII text, which the client may display on an available output device.

The code for this option is 56, and its minimum length is 1.

Code	Length	Text
56	n	c1 c2 [...]

E.3.8 Option 57: Maximum DHCP Message Size

This option specifies the maximum length of a DHCP message that a system is willing to accept. The length is specified as an unsigned 16-bit integer. A client may use the maximum DHCP message size option in DHCPDISCOVER or DHCPREQUEST messages, but should not use the option in DHCPDECLINE messages.

The code for this option is 57, and its length is 2. The minimum legal value is 576 octets.

Code	Length	Length
57	2	11 12

E.3.9 Option 58: Renewal (T1) Time Value

This option specifies the time interval from address assignment until the client transitions to the RENEWING state. The value is in seconds and is specified as a 32-bit unsigned integer.

The code for this option is 58, and its length is 4.

Code	Length	T1 Interval			
58	4	t1	t2	t3	t4

E.3.10 Option 59: Rebinding (T2) Time Value

This option specifies the time interval from address assignment until the client transitions to the REBINDING state. The value is in seconds and is specified as a 32-bit unsigned integer.

The code for this option is 59, and its length is 4.

Code	Length	T2 Interval			
59	4	t1	t2	t3	t4

E.3.11 Option 60: Vendor Class Identifier

This option is used by DHCP clients to optionally identify the vendor type and configuration of a DHCP client. The information is a string of (n) octets, interpreted by servers. Vendors may choose to define specific vendor class identifiers to convey particular configuration or other identification information about a client. For example, the identifier may include the client's hardware configuration. Servers not equipped to interpret the class-specific information sent by a client must ignore it, although it may be reported. Servers that respond should only use option 43 to return the vendor-specific information to the client.

The code for this option is 60, and its minimum length is 1.

Code	Length	Vendor Class Identifier		
60	n	i1	i2	[...]

E.3.12 Option 61: Client Identifier

This option is used by DHCP clients to specify their own unique identifiers. DHCP servers use this value to index their database of address bindings. This value is expected to be unique for all clients in an administrative domain.

Identifiers should be treated as opaque objects by DHCP servers. The client identifier may consist of type-value pairs similar to "htype"/"chaddr" fields. For instance, it may consist of a hardware type and hardware address. In this

case, the type field should be a defined ARP hardware type. A hardware type of 0 (zero) should be used when the value field contains an identifier other than a hardware address (for instance, a fully qualified domain name).

For correct identification of clients, each client's identifier must be unique among the identifiers used on the subnet to which the client is attached. Vendors and system administrators are responsible for choosing identifiers that meet this requirement for uniqueness.

The code for this option is 61, and its minimum length is 2.

Code	Length	Type	Client Identifier
61	n	t1	i1 i2 [...]

E.3.13 Option 66: TFTP Server Name

This option is used to identify a TFTP server when the "sname" field in the DHCP header has been used for DHCP options.

The code for this option is 66, and its minimum length is 1.

Code	Length	TFTP server
66	n	c1 c2 c3 [...]

E.3.14 Option 67: Boot File Name

This option is used to identify a boot file when the "file" field in the DHCP header has been used for DHCP options.

The code for this option is 67, and its minimum length is 1.

Code	Length	Boot File Name
67	n	c1 c2 c3 [...]

E.4 Unofficial DHCP Options

Several additional options, while not yet part of the official RFC 2132 standard, have achieved some degree of popularity and are in common use. If you are interested in the latest inventory of unofficially assigned DHCP options, the list is available on the Internet at:

<ftp://ftp.isi.edu/in-notes/iana/assignments>

Option	Description
62	NetWare/IP Domain Name
63	NetWare/IP Suboptions
77	User Class
78	Directory Agent Information
79	Service Location Agent Scope
80	Naming Authority
81	Client Fully Qualified Domain Name
82	Agent Circuit ID
83	Agent Remote ID
84	Agent Subnet Mask
85	Novell Directory Services Servers
86	Novell Directory Services Tree Name
87	Novell Directory Services Context
88	IEEE 1003.1 POSIX Timezone
89	Fully Qualified Domain Name
90	Authentication
91	Banyan Vines TCP/IP Server
92	Server Selection
93	Client System Architecture
94	Client Network Device Interface
95	Lightweight Directory Access Protocol (LDAP)
96	IPv6 Transitions
97	UUID/GUID-based Client Identifier
100	Printer Name
101	Multicast Scope
102	Start Time
103	Multicast Time-to-Live
104	Multicast Block Size
105	Client Port
106	Cookie
107	Multicast Scope List
108	Swap Path
109	Autonomous System Number (ASN)
110	IPX Compatibility
111	Served IP Range
112	NetInfo Parent Server Address
113	NetInfo Parent Server Tag
114	Universal Resource Locator (URL)
115	DHCP Failover Protocol
126	Extension
127	Extension

By convention, OS/2 Warp (and likely other IBM products) may use additional DHCP options in the 200 to 208 range. You may wish to use these same option numbers for the same purposes, even on non-IBM platforms.

200	Default LPR Printer
201	Default Gopher Server
202	Default WWW Home Page
203	Default WWW Proxy Gateway
204	Default WWW News Server
205	Default SOCKS Server
206	NFS Mount Points
207	Default X Font Servers
208	X Display Manager Servers

E.5 Options Supported by Popular Operating Systems

This section provides information about strengths and weaknesses some DHCP servers and DHCP clients encounter.

E.5.1 Servers

OS/2 Warp Server, AIX, Shadow IPserver, and Windows NT Server can serve any and all DHCP options to clients. However, it appears that it is not possible to alter Option 1 in at least some releases of Windows NT Server in order to assign a subnet mask to a client which is not identical to the server's own subnet mask.

OS/2 Warp Server and AIX tend to provide the most flexibility in defining address pools.

E.5.2 Clients

OS/2 Warp and WorkSpace On-Demand support all DHCP options, and DHCP option handlers are provided for most of the standard DHCP options. Others can be interpreted and acted upon by the client using custom option handlers. (See 7.2, "Roaming Users" on page 284 for some examples.)

Windows 95 and Windows NT support a more limited set of options, notably 1, 3, 6, 44, 46, and 47. Third-party products may be required to intercept and handle less commonly used DHCP options.

Appendix F. Special Notices

This publication is intended to help small TCP/IP network operators establish a solid foundation for future growth without undue expense. Enterprise network staff will appreciate the sections exploring high-end technologies, including Network TeleSystems' Shadow IPserver, IBM Communications Server, and UNIX platforms. All network managers should find the authors' advice on security well worth reading.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594 USA.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM ("vendor") products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no

guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

Any performance data contained in this document was determined in a controlled environment, and therefore, the results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

The following document contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples contain the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

IBM ®	IBM Network Station
OS/2	WorkSpace On-Demand
Lotus Domino	PC-DOS and PC-DOS 2000
Lotus Notes	Dynamic IP
Lotus Domino Go	IBM eNetwork Communications Server
AIX	NetFinity
ThinkPad	ValuePoint
TME 10	DB2

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc.

Java and HotJava are trademarks of Sun Microsystems, Incorporated.

Microsoft, Windows, Windows NT, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

Pentium, MMX, ProShare, LANDesk, and ActionMedia are trademarks or registered trademarks of Intel Corporation in the U.S. and other countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Other company, product, and service names may be trademarks or service marks of others.

Appendix G. Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

G.1 International Technical Support Organization Publications

For information on ordering these ITSO publications see “How To Get ITSO Redbooks” on page 449 or visit <http://www.redbooks.ibm.com> on the Internet.

- *Network Clients for OS/2 Warp Server: OS/2 Warp 4, DOS/Windows, Windows 95/NT, and Apple Macintosh*, SG24-2009
- *OS/2 Warp Server, Windows NT, and NetWare: A Network Operating System Study*, SG24-4786
- *Learning Practical TCP/IP for AIX 3.2/V4.1 Users: Hints and Tips for Debugging and Tuning*, SG24-4381
- *IBM WorkSpace On-Demand Handbook*, SG24-2028
- *AS/400 TCP/IP Autoconfiguration: DNS and DHCP Support*, SG24-5147
- *A Comprehensive Guide to Virtual Private Networks, Vol.1: IBM Firewall, Server and Client Solutions*, SG24-5201
- *Inside OS/2 Warp Server, Volume 1: Exploring the Core Components*, SG24-4786
- *TCP/IP Implementation in an OS/2 Warp Environment*, SG24-4730
- *TCP/IP Tutorial and Technical Overview (Fifth Edition)*, GG24-3376
- *Understanding Performance Tuning Theory for IBM OS/2 LAN Server*, GG24-4430
- *Using IBM Firewall 3.1 for AIX*, SG24-2577

G.2 Redbooks on CD-ROMs

IBM ITSO publications are also available on CD-ROM. **Order a subscription** and receive updates 2 to 4 times a year at significant savings.

CD-ROM Title	Subscription Number	Collection Kit Number
System/390 Redbooks Collection	SBOF-7201	SK2T-2177
Networking and Systems Management Redbooks Collection	SBOF-7370	SK2T-6022
Transaction Processing and Data Management Redbook	SBOF-7240	SK2T-8038

CD-ROM Title	Subscription Number	Collection Kit Number
Lotus Redbooks Collection	SBOF-6899	SK2T-8039
Tivoli Redbook Collection	SBOF-6898	SK2T-8044
AS/400 Redbooks Collection	SBOF-7270	SK2T-2849
RS/6000 Redbooks Collection (HTML and BookManager)	SBOF-7230	SK2T-8040
RS/6000 Redbooks Collection (Postscript)	SBOF-7205	SK2T-8041
RS/6000 Redbooks Collection (PDF Format)	SBOF-8700	SK2T-8043
Application Development Redbooks Collection	SBOF-7290	SK2T-8037
IBM NetFinity and PC Server Software Redbooks Collection, HTML and PDF	SBOF-8699	SK2T-8046

G.3 Other Publications

These publications are also relevant as further information sources:

- *Internetworking with TCP/IP, Volume I, Principles, Protocols and Architecture*, third edition, Prentice-Hall, Inc., 1995, by Douglas E. Comer; ISBN 0-13-216987-8.
- *TCP/IP Tutorial and Technical Overview*, fifth edition, IBM Corp., 1995, GG24-3376-04, and Prentice-Hall, Inc., 1995, by Eamon Murphy, Steve Hayes, Matthias Enders; ISBN 0-13-460858-5.
- *IPng and the TCP/IP Protocols*, John Wiley & Sons, Inc., 1996, by Stephen A. Thomas; ISBN 0-471-13088-5.
- *Communications for Cooperating Systems - OSI, SNA and TCP/IP*, Addison-Wesley Publishing Company, Inc., 1992, by R.J. Cypser; ISBN 0-201-50775-7.
- *DNS and BIND*, 2nd Edition; O'Reilly & Associates; Albitz and Liu; ISBN: 1-56592-236-0.
Note: A third edition of this book will be available in September 1998.
- *Sendmail (Nutsell Handbook)*, 2nd Edition (January 1997); O'Reilly & Associates; Costales and Allman; ISBN: 1-56592-222-0.
- *Internetworking with Microsoft TCP/IP in Microsoft Windows NT 4.0*, Microsoft 0871B

G.4 Request for Comments (RFC)

There are more than 2,200 RFCs today. For those readers who want to keep up-to-date with the latest advances in TCP/IP, the ever-increasing

number of RFCs and Internet Drafts (IDs), published by the non-profit Internet Engineering Task Force, are the best sources. RFCs can be viewed on the Internet at:

<http://www.isi.edu/rfc-editor>

How To Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO publications, CD-ROMs, workshops, and residencies. A form for ordering books and CD-ROMs is also provided.

This information was current at the time of publication but is always subject to change. The latest information may be found at <http://www.redbooks.ibm.com> on the Internet.

How IBM Employees Can Get ITSO Redbooks

Employees may request ITSO deliverables (redbooks, BookManager files, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **PUBORDER** – to order printed copies in United States
- **GOPHER link to the Internet** – type GOPHER WTSCPOK.ITSO.IBM.COM
- **Tools Disks**

To get LIST3820s of redbooks, type one of the following commands:

```
TOOLS SENDTO EHONE4 TOOLS2 REDPRINT GET SG24xxxx PACKAGE
TOOLS SENDTO CANVM2 TOOLS REDPRINT GET SG24xxxx PACKAGE (Canadian users only)
```

To get lists of redbooks:

```
TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET ITSOCAT TXT
```

To register for information on workshops, residencies, and redbooks:

```
TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ITSOREGI 199x
```

For a list of product area specialists in the ITSO:

```
TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ORGCARD PACKAGE
```

- **Redbooks Web Site on the IBM Intranet**

<http://w3.itso.ibm.com/redbooks>

- **IBM Direct Publications Catalog on the World Wide Web**

<http://www.elink.ibm.link.ibm.com/pbl/pbl>

IBM employees may obtain LIST3820s of redbooks from this page.

- **REDBOOKS category on INEWS**
- **Online** – send orders to: USIB6FPL at IBMMAIL or DKIBMBSH at IBMMAIL
- **Internet Listserver**

With an Internet e-mail address anyone can subscribe to the IBM Announcement Listserver. To initiate the service, send an e-mail note to announce@webster.ibm.link.ibm.com with the keyword `subscribe` in the body of the note (leave the subject line blank). A category form and detailed instructions will be sent to you.

How Customers Can Get ITSO Redbooks

Customers may request ITSO deliverables (redbooks, BookManager files, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **Online Orders** (Do not send credit card information over Internet e-mail!) – send orders to:

	IBMMAIL	Internet
In United States	USIB6FPL	usib6fpl@ibmmail.com
In Canada	CAIBMBKZ	lmannix@vnet.ibm.com
Outside North America	DKIBMBSH	bookshop@dk.ibm.com

- **Telephone Orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU

Outside North America	(long distance charges apply)
(+45) 4810-1320 - Danish	(+45) 4810-1020 - German
(+45) 4810-1420 - Dutch	(+45) 4810-1620 - Italian
(+45) 4810-1540 - English	(+45) 4810-1270 - Norwegian
(+45) 4810-1670 - Finnish	(+45) 4810-1120 - Spanish
(+45) 4810-1220 - French	(+45) 4810-1170 - Swedish

- **Mail Orders** – send orders to:

IBM Publications	IBM Publications	IBM Direct Services
Publications Customer Support	144-4th Avenue, S.W.	Sortemosevej 21
P.O. Box 29570	Calgary, Alberta T2P 3N5	DK-3450 Allerød
Raleigh, NC 27626-0570	Canada	Denmark
USA		

- **Fax** – send orders to:

United States (toll free)	1-800-445-9269
Canada	1-800-267-4455
Outside North America	(+45) 48 14 2207 (long distance charge)

- **Fax Back Service**

1-800-IBM-4FAX (United States) or (+1) 408 256 5422 (Outside USA) – request:

Document # 4421: Abstracts for New Redbooks
Document # 4422: IBM Redbooks
Document # 4420: Redbooks for Last Six Months

- **Direct Services** – send note to softwareshop@vnet.ibm.com

- **On the World Wide Web**

Redbooks Web Site	http://www.redbooks.ibm.com
IBM Direct Publications Catalog	http://www.elink.ibm.link.ibm.com/pbl/pbl

- **Internet Listserver**

With an Internet e-mail address, anyone can subscribe to the IBM Announcement Listserver. To initiate the service, send an e-mail note to announce@webster.ibm.link.ibm.com with the keyword `subscribe` in the body of the note (leave the subject line blank).

IBM Publications Order Form

Please send me the following:

Title	Order Number	Quantity

First Name	Last Name
------------	-----------

Company

Address

City	Postal Code	Country
------	-------------	---------

Telephone Number	Fax Number	VAT Number
------------------	------------	------------

<input type="checkbox"/> Invoice to IBM customer no.:	
---	--

<input type="checkbox"/> Credit card number:	
--	--

Credit card expiration date	Card issued to	Signature
-----------------------------	----------------	-----------

We accept American Express, Diners Club, Eurocard, MasterCard, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

Thank you for your order!

Glossary and Abbreviations

ACM. Association for Computing Machinery. Professional organization of computing academics and other experts. See <http://www.acm.org> on the Internet.

ADSL. Asynchronous Digital Subscriber Line. Method of providing high-speed data transfer over standard copper telephone wiring. Typically offers higher performance than ISDN.

AFS. Andrew File System. One of three popular network file systems, originating with UNIX, to provide file and disk sharing across a TCP/IP network. See also DFS and NFS.

AIX. IBM's UNIX-compatible operating system for the RS/6000 line of workstations and servers.

ANSI. American National Standards Institute. Group which promulgates several different technical standards, including the popular ANSI terminal emulation standard.

APAR. A documented problem in an IBM product, normally one for which a patch or other fix is available.

AppleTalk. Network protocol used primarily to connect computers and devices produced by Apple Computer. AppleTalk is based on the ISO/OSI Reference Model and incorporates the SPX protocol.

ARP. Address Resolution Protocol. A protocol used between routers and nodes to determine the MAC or OSI physical layer address when the Network layer (IP) address is known.

AS/400. Application System/400. IBM's line of midrange computing systems, popular for line-of-business applications.

ASCII. American Standard Code for Information Interchange. A standard correspondence between alphanumeric and control characters, and (normally) 7-bit binary values (0 to 127). For example, code 65 is defined as the uppercase letter "A" in the ASCII table. Facilitates communication between different computer systems. Other common character set

definitions include Unicode (16-bit) and EBCDIC (8-bit).

ATM. Asynchronous Transfer Mode. It is a high-speed (155 - 162 Mbps) communications transport facility capable of carrying voice, data, and video signaling.

Binary. Having two components or possible states. Usually represented by a code of zeros and ones.

Boot Manager. Included with IBM's OS/2 Warp products and PowerQuest's Partition Magic software. Presents a menu on startup which allows a PC user to choose which operating system (OS/2 Warp, Windows 95, DOS, Windows NT, and so on) to run among those installed.

BootP. Bootstrap Protocol. An early method of providing IP address information from a central server based on a client's network adapter address. See also DHCP.

Bridge. A network "relay" which reads, buffers, and forwards datagrams from one part of a network to another. As a result, the two parts of a network act as one. Bridges operate at the Datalink layer of the OSI Model, or more precisely, at the Media Access Control (MAC) sublayer. See also Filter.

Broadcast. A transmission of a message (such as a packet or frame) sent to all nodes on a network rather than to a specific station.

Brouter. A single device which acts as both a bridge and a router.

BSD. Berkeley Standard Distribution, a version of UNIX.

Byte. Short for "binary digit eight". A unit of information consisting of usually eight bits. One byte is sometimes referred as one octet.

Cable Modem. A device used to allow a computer to communicate over cable television coaxial wiring, usually to provide high-speed Internet service.

CGI. Common Gateway Interface, a standard mechanism used to pass data collected by a Web server to a server-based application and receive back any results, often used for "intelligent" Web pages, such as order entry forms and credit card processing.

Chooser. The application used on the Macintosh to access AppleShare network services.

CIFS. Common Internet File System protocol.

CPSR. Computer Professionals for Social Responsibility. See <http://www.cpsr.org> on the Internet.

CSD. Corrective Service Diskette(s). Collection of program fixes issued periodically for IBM software products. Also known as Service Packs.

Daemon. A task which runs in the background, usually to provide some kind of network service to clients. For example, FTPD stands for FTP Daemon.

Datagram. A packet of information transmitted across a computer network consisting of a header and one or more bytes of actual data requiring no response or acknowledgement.

DDNS. Dynamic Domain Name System. See also DNS.

DCE. Distributed Computing Environment. The Open Software Foundation's TCP/IP-based protocols for security, DFS, and distributed applications (RPC).

DFS. Distributed File System. One of three popular network file systems, originating with UNIX, to provide file and disk sharing across a TCP/IP network. Part of the Open Software Foundation's DCE. See also AFS and NFS.

DHCP. The Dynamic Host Configuration Protocol server provides IP addresses and IP configuration information to IP address-requesting workstations.

DLS. DOS LAN Services. IBM product for PCs running DOS and/or Windows 3.x. Provides network connection capabilities for file and printer sharing. Also provides network protocol support, including DHCP and basic TCP/IP services. Included with Warp Server.

Domain. In the Internet, a domain is a part of the naming hierarchy. The domain name is a sequence of names (separated by periods) that identify host sites. The leftmost part of a domain name is the most specific, and the part on the right is the most general.

DNS. The Domain Name System (DNS) is a hierarchical, distributed method of organizing systems and network names on the Internet. DNS administratively groups hosts (systems) into a hierarchy of authority that allows addressing and other information to be widely distributed and maintained. A big advantage of DNS is that using it eliminates dependence on a centrally-maintained file that maps host names to IP addresses.

DOS. Disk Operating System. The software programs that control the operation of the computer and the movement of information throughout the computer system. Although operating systems known as "DOS" have been produced for other systems, most people know DOS as one of the popular operating systems for the PC. First introduced in 1981 by Microsoft and IBM, the newest version is called IBM PC-DOS 2000. See

<http://www.software.ibm.com/os/dos> for more information.

Duplex. Pertaining to simultaneous two-way independent data transmission in both directions (as in full duplex).

E-mail. Electronic mail (e-mail) is the most popular Internet application, and the driving force behind Internet's rapid growth.

Ethernet. LAN datalink protocol developed by a consortium of vendors, later standardized as IEEE 802.3 with a few modifications. (For many applications, users have not adopted all the IEEE 802.3 differences.) Ethernet 802.3 can now be run on two types of coaxial cable as well as on multi-mode fiber and unshielded twisted-pair cables. The raw rate of data transmission is typically 10 megabits/second.

Extranet. Using Internet technologies (such as Web, Java, and TCP/IP) to link an internal company network with networks operated by

vendors, suppliers, company partners, and others, usually with at least some security. See also Internet and Intranet.

Fast Ethernet. A nickname for the 100 Mbps version of IEEE 802.3.

FDDI. Fiber Distributed Data Interface (100 Mbps fiber optic LAN).

File. A sequence of bytes stored on a secondary storage medium, such as floppy disk or hard disk.

Filter. A network device or agent (program) designed to separate data, signals, or material in accordance with specified criteria.

Firewall. Generic term for a sophisticated filter designed to connect a private network (Intranet) to a public network (Internet) in a secure way. See also SOCKS and Proxy.

FTP. The File Transfer Protocol is part of the TCP/IP protocol suite that is used to transfer files between any two computers, provided they support FTP. The two computers do not have to be running the same operating system.

FQDN. Fully Qualified Domain Name. Includes hostname plus domain name.

Gigabit Ethernet. Refers to Ethernet technologies promising 1000 Mbps raw transmission speed.

GUI. Graphical User Interface. A GUI uses graphic representations of commands and/or a menu format to display commands that the user may execute with a mouse or similar device.

Host. Any computer attached to a TCP/IP network that is remotely accessible through an address.

HPFS. High Performance File System, part of OS/2 Warp. The 32-bit version of this file system is called HPFS386.

HTML. The HyperText Markup Language is the standard language that the Web uses for creating and recognizing hypermedia documents. Web documents are most often written in HTML and normally have an .html or .htm extension.

HTTP. HyperText Transport Protocol, the method used by Web browsers to retrieve Web pages from a Web server across a TCP/IP network.

Hub. A central point, or terminus, for network connections or wiring.

Hyperlink. Words, phrases, images, or characters highlighted in bold or underlined indicate connections in a given document to information within another document.

Hypermedia. Hypermedia is the name for richly formatted documents containing a variety of information types, such as textual, image, video, and audio. These information types are easily found through hyperlinks.

Hypertext. Hypertext allows users to move from one site or place in a document to another. Hypertext links in World Wide Web documents link the user from terms in one document to the site referenced in the original document.

IBM. International Business Machines.

ICMP. The Internet Control Message Protocol is used for error reporting and recovery, and is a required component of any IP implementation. Described by RFC 792.

Icon. A graphical picture used to represent an application, folder, file, disk drive, or printer.

IDE. Integrated Drive Electronics. This is a standard interface for a hard disk drive or CD-ROM drive. An IDE bus can be identified by its 40-pin connector, as opposed to the 50-pin connector of a SCSI bus.

IEEE. Institute of Electrical and Electronic Engineers. A professional ANSI-accredited body of scientists and engineers based in the United States. See <http://www.ieee.org> on the Internet.

IEEE 802. The set of IEEE standards for the definition of LAN protocols.

IEEE 802.2. An IEEE standard describing the parts of all IEEE LAN datalink protocols that are common.

IEEE 802.3. An IEEE standard for LANs (an improved version of Ethernet). See also Ethernet.

IEEE 802.5. An IEEE standard for token-ring LANs. There are three types: 4 Mbps, 16 Mbps, and 100 Mbps. See also Token-Ring.

IETF. Internet Engineering Task Force, a group responsible for defining protocols in the TCP/IP family for use on the Internet. See <http://www.ietf.org> on the Internet.

IFS. Installable File System, a program used to manage file storage on disks and other storage devices. The program can be added or removed from an operating system to change its file storage behavior. For example, installable file systems on OS/2 Warp include HPFS, HPFS386, NFS, CD-ROM, and others.

IGMP. Internet Group Management Protocol, maps IP addresses to hardware addresses on a network. Also allows routers to check hosts to see if they are interested in participating in multicasts.

IMAP. Internet Message Access Protocol, used primarily for TCP/IP-based e-mail applications.

Internet. The worldwide public network linking computers using TCP/IP, Java, and other standard networking technologies. Began in the U.S. as a result of collaboration between the Department of Defense, research institutions, and universities. See also Extranet and Intranet.

Intranet. A private network established entirely within a company or other organization based on Internet technologies. See also Extranet and Internet.

IP. Internet Protocol, the basic protocol of TCP/IP and of the Internet. IP is the OSI layer 3 routed protocol used to transmit packetized information on a TCP/IP network.

IP Address. Each host in the network is assigned a unique IP address for each network connection (installed network adapters). The IP address is used to identify packet source and destination host.

IPv6. Internet Protocol Version 6, described by RFCs 1883 through 1887.

IPX. A network protocol developed by Novell for use with its NetWare products.

IRC. Internet Relay Chat. A mechanism for providing live messages across a TCP/IP network, to allow individuals to converse in groups.

ISDN. Integrated Services Digital Network. A digital telephone service used for high speed Internet access and voice services. Delivered over standard copper telephone wiring. See also ADSL.

ISP. Internet Service Providers are companies that provide an Internet connection for educational institutions, individuals, companies, and organizations.

ITSO. International Technical Support Organization, part of IBM Corp. See <http://www.redbooks.ibm.com> for more information.

Java. Java is a general-purpose concurrent object-oriented programming language. Its syntax is similar to C and C++, but it omits many of the features that make C and C++ complex, confusing, and unsafe. Java was initially developed to address the problems of building software for networked consumer devices. It was designed to support multiple host architectures and to allow secure delivery of software components. To meet these requirements, compiled Java code had to survive transport across networks, operate on any client, and assure the client that it was safe to run. With Java as the extension language to HTML, a Web browser is no longer limited to a fixed set of capabilities. Programmers can write an applet once and it will run on any machine, anywhere.

JVM. The Java Virtual Machine is the cornerstone of Sun's Java programming language. It is the component of the Java technology responsible for Java's cross-platform delivery, the small size of compiled code, and Java's ability to protect users from malicious programs. It is an environment provided for any conventional operating system (such as UNIX, OS/2 Warp, or Windows) designed to run Java applications and applets. (Native Java machines run Java bytecode directly, without interpretation or translation.) The JVM knows nothing of the Java programming language, only of a particular

file format, the *class* file format. A *class* file contains JVM instructions (or *bytecodes*) and a symbol table, as well as other ancillary information. Java is a trademark of Sun Microsystems. Consequently a "Java" virtual machine has passed JavaSoft compliance testing. See also applet.

LAN. A Local Area Network is a group of computers running specialized communications software, and joined through an external data path. A LAN will cover a small geographic area, usually no larger than a single building. The computers have a direct high-speed connection between all workstations and servers, and share hardware resources and data files.

Linux. UNIX-compatible operating system originally developed by Linus Torvalds.

LPD. Line Printer Daemon, a server-based application designed to act as a print server on TCP/IP networks.

LPR. A TCP/IP-based application which can forward print jobs to any LPD server.

LSM. IBM LAN Server for Macintosh.

MAC. Media Access Control, pertaining to network adapters.

MD5. Message Digest 5. See RSA MD5.

Modem. Modem is an abbreviation for modulator/demodulator. A modem is a peripheral device that permits all sort of computers to receive and transmit data in digital format across voice-oriented communications links, such as telephone lines.

MSS. Maximum Segment Size, the maximum size of a TCP segment.

MTU. Maximum Transmission Unit, the size of an IP datagram.

Multicast. Network traffic intended to be received by more than one system (but not necessarily every system, as would a broadcast).

NBDD. NetBIOS Datagram Distributor.

NBNS. NetBIOS Name Server. Resolves NetBIOS names to IP addresses eliminating typical NetBIOS broadcast traffic.

NC. Network Computer or Network Computing.

NDIS. Network Driver Interface Specification, a standard used to create drivers for network adapters.

NetBEUI. NetBIOS Extended User Interface. This is a non-routable transport protocol written to the NetBIOS interface. It usually implies NetBIOS over NetBEUI.

NetBIOS. NetBIOS is a standard programming interface (API) for the development of distributed applications. It can run over different transport protocols, such as NetBEUI, TCP/IP, or IPX/SPX.

NETBT. Microsoft's term for NetBIOS over TCP/IP. IBM calls it TCPBEUI.

NIC. 1.) Network Interface Card (network adapter). 2.) Network Information Center. The most famous one on the Internet is the InterNIC, which is where new domain names are registered.

NFS. Network File System. One of three popular network file systems, originating with UNIX, to provide file and disk sharing across a TCP/IP network. See also AFS and DFS.

NNTP. The Network News Transport Protocol defines the distribution, inquiry, and retrieval of news articles on the Internet from TCP/IP sites.

NT. Microsoft operating system (Windows NT).

NTP. Network Time Protocol. It is the protocol that assures accurate local time keeping with reference to radio and atomic clocks located on the Internet.

NVRAM. Nonvolatile Random Access Memory. Normally information stored in RAM will be lost without continuous electrical power, but NVRAM can hold such information intact even without external power. The IBM Network Station contains some NVRAM used to store boot parameters.

NVT. Network Virtual Terminal, as in NVT ASCII. Defined by Military Standard 1782, which specifies a standard form for ASCII representation and for certain other characteristics of a simple terminal. See also ASCII.

ODI. Open Datalink Interface, a driver specification used primarily with Novell networks.

OS/2. Operating System/2. IBM operating system for the PC.

OSI. Open Systems Interconnection. To support international standardization of network terminology and protocols, the International Standards Organization (ISO) proposed a reference model of open systems interconnection.

OSI Reference Model. The OSI model provides a standard means of describing the data flow in a network and how it is managed:

- 7 Application Layer
- 6 Presentation Layer
- 5 Session Layer
- 4 Transport Layer
- 3 Network Layer
- 2 Data Link Layer
- 1 Physical Layer

OSI layer 1. The Physical Layer. It is the lowest of the seven defined layers of the generalized network architecture. It defines the transmission of bits over a communication channel, ensuring that 1s and 0s are recognized as such.

OSI layer 2. The Data Link Layer. It provides methodologies for transforming the new physical layer link into a channel that appears free of errors to the network layer (the next higher layer). The data link layer accomplishes this by splitting the input or data stream provided in the physical layer into data frames that are transmitted sequentially as messages and by processing the acknowledgement (ACK) frames sent back over the channel by the receiver.

OSI layer 3. The Network Layer. It accepts messages of data frames from the transmitting host, converts the messages to packets, and routes the packets to their destination.

OSI layer 4. The Transport Layer. It accepts data from the session layer (the next layer up, which is the human user's interface to the network), splits this data into smaller units, passes these units down to the network layer, and ensures that all the pieces arrive at the destination in correct

order. The transport layer is a true end-to-end process.

OSI layer 5. The Session Layer. It is the user's interface into the network through which the user establishes a connection with a process on another distant machine. Once the connection is established, the session layer manages the end-to-end dialog in an orderly manner, supplementing the application-oriented user functions to the data units provided by the transport layer.

OSI layer 6. The Presentation Layer. Its protocols format the data to meet the needs of different computers, terminals, or presentation media in the user's end-to-end communications. The protocols at this layer may also provide data encryption for security purposes in transmission over networks, or data compression for efficiency and economy.

OSI layer 7. The Application Layer. It specifies the protocols for the user's intended interaction with the distant computer, including such applications as database access, document interchange, or financial transactions.

OSPF. Open Shortest Path First routing protocol, which is the fastest and most efficient routing protocol. Routers are calculated based on real performance and are dynamically adjusted if a route becomes congested. See also RIP.

Packet. A unit of data transmitted at the OSI network layer; or any addressed segment of data transmitted on a network.

Peer. A system connected to a network which communicates more or less on equal terms with another system on the network, usually providing basic file and print sharing. Clients, unlike peers, generally must depend on servers for vital functions. In other words, peers can alternately behave as both clients and servers in a network.

Ping. Packet Internet Groper. A program used to test reachability of destinations by sending them an ICMP echo request and waiting for a reply.

PM. Presentation Manager. Graphical subsystem for displaying windows, icons, and other GUI

components on IBM's OS/2 Warp operating system.

POP. The Post Office Protocol is one of mail exchange protocols.

PPP. Point-to-Point Protocol. The successor of the SLIP protocol, PPP allows a computer to use a regular telephone line and a modem to make IP connections. PPP can also carry other routable protocols such as IPX.

Protocol. The "rules" by which two network devices trade information in order to communicate. Must include rules about a lot of mundane detail as well as rules about how to recover from a lot of unusual communication problems. Thus, the rules can be quite complicated.

Proxy. An agent which can forward requests to another part of the network on behalf of clients, returning any results. Most often used to allow Web browsers to access public Web sites beyond a firewall.

PTF. Program Temporary Fix, issued by IBM to correct a defect in an IBM software product.

QOS. Quality Of Service. Typically a specified bandwidth or timeliness of delivery of packets. *Controlled load* reservation has a specified data rate, whereas *guaranteed* reservation additionally has an upper bound on delivery time.

RARP. The Reverse Address Resolution Protocol is used to map the MAC, or hardware address, to a host's IP, or software address.

RedHat. One of two major vendors supplying commercial versions of Linux. See also Slackware.

Requester. A LAN client which requests file and print services from a server such as OS/2 Warp Server.

REXX. Restructured EXTended eXecutor language. Computer programming and scripting language developed by IBM's Mike Cowlishaw. Provided as a standard part of most IBM operating systems. Available for nearly all operating systems.

RFC. Request for Comments. Refers to documents published by the Internet Engineering Task Force (IETF) which have achieved at least some degree of standardization. RFCs describe various TCP/IP protocol standards.

RIP. Routing Information Protocol. This protocol allows routers to exchange routing details on a network. Using RIP, routers can create and maintain a database, or routing table, of current information. Workstations can query the nearest router to determine the fastest route to a distant network by broadcasting a RIP request package. Routers send periodic RIP broadcast packets with current information to keep all routers on the internetwork synchronized.

RIPL. Remote Initial Program Load. Method of obtaining boot software across a network to start a PC or other computer, often one without a disk drive.

Router. A network "relay" that uses a protocol beyond the datalink protocol to route traffic between LANs and other network links.

Routing Protocol. A protocol sent between routers by which routers exchange information on how to route to various parts of the network. TCP/IP includes several routing protocols such as RIP, EGP, BGP, OSPF, and dual IS-IS.

RPC. Remote Procedure Call.

RSA MD5. Rivest-Shamir-Aleman algorithm Message Digest 5..

RSVP. Resource ReSerVation Protocol. A protocol to reserve quality of service (QOS) for a data stream. See 10.5.2, "RSVP" on page 367.

SCSI. Small Computer Systems Interface. A high-speed interface bus used for disk drives, tape drives, scanners, printers, CD-ROM drives, digital cameras, and other devices. Available in several versions including SCSI-I, SCSI-II (Fast SCSI), Wide (16-bit data path) and UltraWide.

Segment. Effectively a TCP datagram.

ServicePaks. See CSD.

Slackware. One of two major vendors supplying commercial versions of Linux. See also RedHat.

SLIP. Serial Line (or Link) Internet Protocol. Method used to provide a TCP/IP connection over serial connections, such as dial-up modems. See also PPP.

SMTP. Simple Mail Transport Protocol, designed to send e-mail across a TCP/IP network.

SNMP. The Simple Network Management Protocol is one of the most comprehensive tools available for TCP/IP network management. It operates through conversations between SNMP agents and management systems. The SNMP management system can collect statistics from and modify configuration parameters on agents.

SOCKS. A technique used on a TCP/IP network to provide secure access through a firewall for most TCP/IP-based applications.

Solaris. Sun's version of the UNIX operating system.

Spam. Unsolicited e-mail, usually sent to huge numbers of people. Also a brand of canned meat, trademark of Hormel.

SPX. A network protocol created by Novell for use with NetWare products.

SrvIFS. A simple file system created by IBM. Used primarily to distribute software to PCs booted from OS/2 Warp diskettes. See also IFS.

Stack. The collection of drivers, APIs, and other files used to implement a network protocol on a computer, as in a TCP/IP protocol stack.

Subnet. The primary reason to divide a network into subnets is network performance and available bandwidth. Without separate networks, each transmission would be broadcast across the entire internetwork, waiting for the destination system to respond. Routers divide, as well as provide communications between the networks.

Switch. A network device used to alter or change network connections, often to help boost performance.

TCP. The Transmission Control Protocol defines connection-oriented, or reliable, transport. See also UDP.

TCPBEUI. IBM's NetBIOS over TCP/IP implementation.

TCP/IP. Transmission Control Protocol/Internet Protocol. The Internet is based on TCP/IP network protocols.

TFTP. The Trivial File Transfer Protocol is part of the TCP/IP protocol suite that is used to transfer files between any two computers, provided they support TFTP. Unlike FTP, TFTP does not use a user ID to ensure proper access rights. However, a password can be set optionally in many TCP/IP implementations.

Token-Ring. People often use the term "Token-Ring" to designate IEEE 802.5. In the more general sense of the term, a token-ring is a type of LAN that has stations wired in a ring, where each station constantly passes a special message (a "token") on to the next. Whichever system has the token can send a message.

UBE. Unsolicited Bulk E-mail. (See Spam)

UCE. Unsolicited Commercial E-mail. (See Spam)

UDP. The User Datagram Protocol is a transport protocol in the Internet suite of protocols. It provides a connection-less, or unreliable, transport.

UNC. Universal Naming Convention. A method used to name network resources on NetBIOS-based file and print sharing systems, such as Warp Server.

UNIX. Operating system originally developed by AT&T's Bell Labs. Historically one of the most popular operating systems used to develop Internet and TCP/IP technologies.

URL. The Universal Resource Locator is the pathname of a document on the Internet. URLs can be absolute or relative. An absolute URL consists of a prefix denoting a "method" (http for Web sites, gopher for gophers, ftp for FTP sites, and so forth). The prefix is followed by a colon and two slashes (://), and an address. The address consists of a domain name followed by a slash and a pathname (or *username@domain name* for mailto). The last part is an optional

anchor which is preceded by a #. The # symbol points to a place within the Web page.

VDM. Virtual DOS Machine. See also VM.

VM. Virtual Machine. A computer system or environment which exists within another, so that the "host" system can run applications written for another system. Examples include Java virtual machines (JVMs) and OS/2 Warp's VDMs (virtual DOS machines). VM also refers to a specific IBM operating system.

VPN. Virtual Private Network. Method used to securely connect intranets in separate locations using the public Internet.

WAN. Wide Area Network. A term for regional, state, country, or worldwide networks developed to link several LANs together.

Windows. Microsoft GUI environment(s).

WINS. Windows Internet Name Service, or WINS, is an automated way of supporting NetBIOS address resolution. It is a modification to NetBIOS Name Service (NBNS).

WWW. The World Wide Web is a recent and fast growing addition to the Internet. The Web unifies many of the existing network tools with hypertext (also called hyperlinks). Some hyperlinks lead to FTP sites, newsgroups, gopher sites, and other Web sites which house additional Web documents. To navigate these Web sites and find links, search engines are available.

Zip. A Zip file contains one or more other files in compressed form. An "unzip" program can be used to extract the original files from the Zip package. Zip files save space and allow multiple files to be transferred in one package.

Index

A

AIX 66, 227, 244, 332, 340, 365
 Configuring DHCP server 244
 DHCP and BOOTP 256
 dhcpsconf 244
 Firewall 324
 HACMP 340
 Start DHCP server (smit dhcpsd) 257
 X-windows 244
Apple Macintosh 28, 47
ARP 346
ARPANET 88
authoritative 92
authoritative name server 96, 97

B

BIND 113, 139
BIND 8.1.1 119, 121
B-Node 181, 182
 enhancing DOS, Windows 3.1, and Windows for
 Workgroups 182
 enhancing OS/2 Warp 188
 enhancing Windows 95/NT clients 185
BOOTP 10, 49, 54, 55, 57, 74, 76, 235, 323, 329,
413
BOOTP relay agent 76, 236
BOUND state 82
broadcast 77, 84, 181, 190, 240, 366

C

caching-only name server 96
CD-ROM
 Contents 411
CIFS 200
Class A 3, 6, 8
Class B 4, 6, 8, 9
Class C 4, 6, 8, 9, 31
Class D 4
Class Determination 5
Class E 4
CNAME
 see also name server
Commands
 APPLET.EXE 382
 APPLETVIEWER.EXE 382

ARP 346
control-panel 56
DADMGUI.CMD 370
DADMIN 29, 54, 153, 339
DDNSAPC.CMD 370
DDNSAPS.CMD 376
DDNSCFG.EXE 127, 133, 159, 160, 169, 175,
176
DDNSZONE 139
DETACH 72
DHCPMON 40, 42
DHCPD.EXE 77, 239
dhcpsconf 245
DHCPSCPC.CMD 370
DHCPSCPS.CMD 376
DHCPD.EXE 338
grep 257
HOST 138, 348, 356
IBMDYNIP.EXE 164, 166, 172
ifconfig 57, 64, 364
IPCONFIG 46, 166
IPFORMAT 346
IPGATE 229, 324
IPREPORT 346
IPTRACE 346
kill 257
LPR 74
MAPNAME.EXE 192
NBTSTAT 200
NBUTIL 182, 184
NETSTAT 155, 347, 353, 354, 364
NSLOOKUP 138, 348, 356, 358
PING 64, 182, 344, 351
refresh 257
REGEDIT2 305
RFCADDR.EXE 191, 292
route 65, 230, 233, 365
SEDISK 386
SETBOOT 337
shutdown 57
smit 257
startsrc 257
startx 56
stopsrc 258
TCPCFG 41, 161
TCPCFG2 39, 50, 123, 158, 168
TEDIT 386

- THINLAPS 386
- touch 65
- TRACERTE 345, 353, 364
- WINIPCFG 44, 167, 172
- computername 180, 320

D

- DADMIN 29

Daemons

- DHCPD 240
- FTPD 22, 332
- INETD 55, 333
- LPD 72
- LPRPORTD 72
- portmapper 55
- REXCD 332
- ROUTED 231
- syslog 50
- TELNETD 22
- TFTP 55

- datagram distributor 204, 214

datagrams

- NetBIOS 190

- DB2 405

- DDNS 12, 68, 87, 94, 112, 281, 310

- Dynamic Presecured Domain (in OS/2 Warp 4) 168

- Dynamic Presecured Domain (in Windows 95) 171

- Dynamic Presecured Mode 116, 130, 312

- Dynamic Secured Mode 116, 124, 312

- ns-updates 127, 133

- ProxyArec 117, 121, 137, 318

- PTR records 129, 136

- DDNS client 94

DDNS Parameter

- DDNSAdministratorClient 144

- deferUpdCnt 144

- incrTime 144

- keyToSec 144

- notify 141

- notify.add 141

- notify.delayTime 141

- notify.remove 141

- notify.retryNumber 141

- notify.retryTime 141

- reverseMapping 144

- safeWrite 142

- sepDynStatic 143

- sigDEL 142

- sigDEL.time 142

- timeSync 142

- timeSync.toSecondaries 142

- ttlSet 143

- ttlSet.value 143

- delegation 91

- DHCP 17, 30, 55, 57, 235, 281

- see also dynamic IP

- DHCP Boot Diskettes 385

- DHCP Monitor 40, 161, 163

- DHCP Options 105

- Option 001 36

- Option 003 32

- Option 006 32, 105, 128, 135

- Option 009 289

- Option 012 320

- Option 015 25, 32, 105, 128, 135

- Option 044 109, 214, 288

- Option 045 214, 288

- Option 046 109, 214, 288

- Option 069 151

- Option 070 152

- Option 071 289

- Option 077 322

- Option 081 320

- Option 114 295, 296

- Option 150 287, 290

- Option 151 287, 290

- Option 192 99, 314

- Option 200 289

- Option 200-208 286

- Option 201 289

- Option 202 289

- Option 205 286, 302, 303

- Options 001-208 413

- DHCP pools 272

- DHCP relay agent 76, 235, 322

- DHCP Server 37, 87, 94

- DHCPACK 81, 82, 85

- DHCPDECLINE 82, 83

- DHCPDISCOVER 76, 78, 81, 83, 235, 343

- DHCPINFORM 85

- DHCPOFFER 78, 79, 80, 83

- DHCPRELEASE 83

- DHCPREQUEST 80, 81, 85

- digital signature 114, 115

- DLS 57, 62, 182

- DNS 11, 12, 30, 87, 93, 96
 - primary 122
- DNS name space 89, 97, 98
- domain 89, 91, 97, 180
 - child 97
 - parent 97
 - primary 122, 125, 131
 - reverse 104
- domain name system 92
- DOMAINSCOPE 192
- Domino Go 369
- Domino Go Webserver 369, 377, 383
- DOS 28, 57
- DOS LAN Services 57
- dynamic IP 17
 - BOOTP Clients 53
 - DHCP 17
 - DHCP Clients 37, 39, 44, 45, 75
 - DHCP Manager 32
 - DHCP Options 31, 34, 52
 - see also DHCP Options
 - DHCP Server 22, 53, 66, 75
 - DHCP server 18, 31
 - DHCPACK 81, 82, 85
 - DHCPDECLINE 82, 83
 - DHCPDISCOVER 76, 78, 81, 83, 235, 343
 - DHCPINFORM 85
 - DHCPOFFER 78, 79, 80, 83
 - DHCPRELEASE 83
 - DHCPREQUEST 80, 81, 85
 - RENEWING state 85

E

- Ethernet 57, 64

F

- Feature Installer 18, 38
- Files
 - AUTOEXEC.BAT 184, 206, 399
 - cache.dns 109
 - CONFIG.SYS 22, 214, 386
 - DDNS.DAT 115, 170
 - dhcp.hme0 65
 - DHCPD.CFG 175, 215, 287, 289, 295, 321
 - DHCPD.CFG 239
 - DHCPD.AR 338
 - DHCPD.CR 338
 - DHCPD.CFG 23, 318

- dhcpsd.cnf 244, 253, 255, 258
- DHCPD.LOG 27
- DNSEXT.CFG 139
- DNSF0000.DOM 145
- DNSF0000.STA 145
- hme0.dhc 66
- hostname.hme0 65
- HOSTS 30, 124, 127, 131, 202
- HTTPD.CNF 377
- IBMLAN.INI 190, 293
- IBMTOKCS.NIF 386
- LANTRAN.LOG 394
- LMHOSTS 185, 202
- masterfile.cnf 252
- MPTSTART.CMD 19
- NAMED.BOOT 145
- NAMED.BT 145
- NBUTIL.LST 184
- NETSCAPE.INI 302, 305, 306
- NETWORK.INI 62, 398
- NSUPDATE.LOG 138
- NTS-CFG-.SAV 260
- NTS-DHC-.SAV 260
- NTS-DNS-.SAV 260
- NTS-NBN-.SAV 260
- NTS-SRVR.CFG 206, 259
- NTS-SRVR.DHC 208, 259
- NTS-SRVR.DNS 208, 259
- NTS-SRVR.NBN 208, 259
- NTS-SRVR.RUR 259
- PROTOCOL.INI 192, 210, 214, 390
- RESOLV2 357
- RFCBCST.LST 190, 292, 295
- RFCNAMES.LST 188, 292
- RHOSTS 30
- SENDMAIL.CF 147, 148, 149
- SENDMAIL.UML 150
- SETUP.CMD 19, 22, 229, 230, 324
- SOCKS.ENV 295, 301
- STARTUP.CMD 292
- TCPSTART.CMD 19, 232
- TFTPAUTH 326
- TFTPPAUTH 53
- WCLLOCAL.INI 283
- firewall 13, 98, 302, 323
- FixPak 18, 305
- forward mapping 101
- forwarder 98
- FQDN 317, 321, 377

G

gateway 9
groupname 180

H

HACMP 340
Hewlett-Packard 28, 69
HINFO
 see also name server
H-Node 181, 213
HOST 348, 356
host 8
hostname 22, 103, 246

I

IBM Network Station 28, 49, 54
IBMWARP_V3.1 287, 288
INIT state 76
internal root 97
Internet 88, 98, 113, 234, 401
 root server 90
intranet 13, 98, 323
IP address 10, 30, 81, 84, 88, 90, 94, 112, 128,
130, 136, 138, 191, 322, 341
IP datagram
 MTU 13
IP masquerading 241
IP packet 243
IPTRACE 346

J

Java 18, 19, 23, 38, 301, 370
JDK 382
JetDirect 69
JIT 370

K

KEY
 see also name server
Keyfile 100

L

LAN 14
LAN interface 0 21, 39, 41
LAN Requester 293
lease 29, 33, 40, 322, 341

lease time 84
 choosing the right one 342
 examples 343
 multiple leases 343
 Rule-of-Thumb 342
leasing server 84
Linux 28, 55
LMHOSTS 186
localhost 22, 377
lookup 190
loopback 5, 22
Lotus Domino 383

M

MAC 52, 315, 318, 322
mail
 routing 90
M-Node 181
MPTS 19, 38, 189, 190, 212
MSBROWSE 180, 209
MTU 13, 363, 364
MX
 see also name server

N

name server 17, 87, 92
 authoritative 96
 caching-only 96
 child 97
 dynamic 94
 firewall 98
 forwarder 98
 master 95
 non-authoritative 96
 parent 97
 parent and child 97
 primary 95
 record types 99
 A 100, 112, 113, 127, 133, 195, 276, 310,
 318, 359
 CNAME 100, 146, 194, 196, 276, 317, 359
 HINFO 100, 194, 359
 KEY 100, 115, 116, 146, 311, 359
 MB 359
 MINFO 359
 MX 99, 146, 147, 153, 359
 NS 99, 110, 146, 359
 PTR 100, 105, 112, 113, 120, 122, 127,

- 128, 133, 136, 258, 315, 318, 359, 406
- RP 99
- RR 100, 310, 318
- SIG 311, 360
- SOA 99, 110, 146, 268, 360
- TXT 99, 145, 360
- UINFO 360
- WKS 360
- root 97
- secondary 95
- static 94, 95
- zone transfer 96
- name space 91, 92, 106
- NBNS 14, 245
 - Configuring DOS, Windows 3.1, and Windows for Workgroups 209
 - Configuring OS/2 Warp 212
 - Configuring Windows 95 211
 - Configuring Windows NT 211
 - see NetBIOS name server 181
- NC 55
- Neighborhood Browser Enabler 222
- NET commands
 - NET START 184
 - NET START BROWSER 224
 - NET START DHCPSEVER 32
 - NET START DNS 103
 - NET START WINS 107
 - NET STOP BROWSER 224
 - NET STOP SERVER 223
- NETAnywhere 281
- NetBIOS name resolution 195
- NetBIOS name server 177, 285, 288
- NetFinity Manager 349
- NetFinity Server 406
- Netscape 18, 19, 38, 156, 244, 295, 375, 410
- NETSTAT 347, 353, 354
- Network Address Translation 242
- NetXRay 75
- Node Status request 195, 198
- Non-RFC-encoded names 195, 200
- NS
 - see also name server
- NSLOOKUP 312, 339, 348, 356, 358
- NSUPDATE 113, 165, 167, 172, 310
- ns-updates 127, 133
- NT
 - see also Windows NT 30
- NVRAM 55

O

- OS/2 Warp 4 28, 36, 37, 39, 158, 161, 168, 284, 296, 385
 - DHCP relay agent 239
 - NBNS Parameter
 - NBDD 213
 - NBNS 213
 - router enablement 229
- OS/2 Warp Server 66, 195, 204, 227, 281
 - Configuring DDNS Dynamic Presecured Mode 130
 - Configuring DDNS Dynamic Secured Mode 124
 - Configuring DHCP 22
 - DDNS Security 115
 - DDNS updates requests 116
 - DHCP Clients 37
 - DHCP server 18
 - Dynamic DNS (DDNS) 112
 - Dynamic DNS Clients 158
 - Dynamic Presecured Mode 121
 - Dynamic Secured Mode 121
 - Installing DHCP server 18
 - Neighborhood Browser Enabler 204, 222
 - NSUPDATE 113
 - ns-updates 127, 133
 - PPP server 280
 - ProxyArec 117, 137
 - Remote IPL 174
 - Routing 21
 - TCP/IP Administrator Password 20
- OS2POPS 147, 156

P

- packets 79
- parent and child name server 97
- Peer Services 58
- PING 82, 344, 364
- P-Node 181, 213
- PPP clients 283
- PPP dialer 281
- PPP Dial-Up 279
- PPP Server 280
 - OS/2 Warp Server 280
- primary name server 95
- private network 98
- protect mode redirector 61
- Protocols

- ARP 2, 13, 79, 82, 346, 347, 351
- BOOTP 49, 53
- FTP 12, 88, 324
- HTTP 12, 324
- ICMP 2, 12, 79, 344, 347
- IEEE 802.2 14
- IGMP 2, 13, 347
- IMAP 12
- IP 2, 12
- IPv6 13
- IPX 14
- NBT 177
- NetBEUI 14, 59, 177, 179, 367
- NetBIOS 14, 58, 177, 179
- NetBIOS over TCP/IP 177
- NNTP 12
- non-routable 177
- OSPF 227
- POP 12
- POP3 147
- PPP 13, 279
- RARP 2, 13, 69
- RIP 227, 231, 234
- routable 177
- RSVP 367
- SLIP 13
- SMB 196, 200
- SMTP 12, 147, 149, 155
- SNMP 12
- TCP 2
- TCP/IP 177
- TCPBEUI 59, 177, 179, 188, 191, 285, 367
 - sockets-send commands 178
 - translation 178
- TFTP 50, 51, 323
- UDP 2, 177, 347
- UDP/IP 177
- ProxyArec 117, 318, 320
 - Option 81 considerations 320
 - Protected 320
 - Standard 320
- PTR
 - see also name server
- public key 115
- public-private key 118

R

- RedHat Linux 55
- Redirector 61
- Remote administration 369
- Remote IPL 67
- RENEWING 85
- resolvers 88
- reverse mapping 101, 104, 133
- REXX 68
- REXX Commands
 - DHCPIBM.CMD 290
 - FAILSAFE.CMD 335
 - GETNBNS.CMD 215, 217, 222
 - MYOFFICE.CMD 290
 - MYWEBMSG.CMD 296, 297
 - NSSOCKS.CMD 302, 303
 - RDADMGUI.CMD 372
 - RDDNSAPC.CMD 370
 - RDHCPSPC.CMD 373
 - SEEMYMSG.CMD 300
 - SETNBNS.CMD 215, 217, 221
- RFC 13, 15
- RFC-encoded name 195, 198
- root name server 97
- router 9, 17, 21, 76, 227
- routing table 10, 228
- RP
 - see also name server
- RR
 - see also name server
- RSA 113, 115, 116, 315
 - 1024-bit key 311
 - 128-bit key 311
 - 56-bit key 311
 - public/private key pair 310
- RSVP 367

S

- Scope 33, 36
- secondary name server 95
- Security 309
 - TFTP 326
- Shadow IPmanager 260
- Shadow IPserver 203, 318, 340
 - Installing and Configuring 206
 - OUTAHERE parameter 208
- SNA 367, 368
- sniffer 75, 80
- SOA
 - see also name server

- sockets 12
- SOCKS 13, 22, 243, 286, 295, 301, 302, 324
- Software Choice 18, 37, 223, 280, 375
- stub resolvers 88
- subdomain 88, 89, 91, 92, 93, 97
- subnet 6, 25, 76, 135, 191, 227, 241
- subnet mask 7, 17, 36
- subnetting 9
- suffix
 - 0x00 193
 - 0x03 193
 - 0x20 193
- Sun JDK 375
- Sun Microsystems 64
- Sun Microsystems Solaris 28
- Supernetting 9

T

- TCP/IP 19
- Telnet 69, 70
- timers
 - T1 84, 85, 341
 - T2 84, 85, 341
- Tivoli 406
- token-ring 57, 64
- TRACERTE 345, 353
- Troubleshooting 350
- T-Sig 119, 121
- TTL 201, 311
- TXT
 - see also name server

U

- unique domain name 90
- UNIX 243, 357
- untrusted network 98
- URL 295

V

- virtual redirector 61
- VPN 13, 325

W

- WAN 342, 366, 367
- Warp Server
 - See OS/2 Warp Server
- Web Server 375

- Windows 3.1 28, 57, 182
- Windows 95 28, 36, 44, 111, 156, 163, 174, 185, 320
 - Modifying the registry 173
- Windows for Workgroups 28, 57, 182
- Windows NT 28, 36, 45, 54, 66, 100, 112, 163, 195, 320
 - Client Support 111
 - Configuring DHCP 32
 - DHCP relay agent 236
 - DHCP server 100
 - DNS (static) 100
 - Firewall 324
 - Installing DHCP 30
 - Installing DNS 102
 - Installing WINS 106
 - Node Status request 196
 - router enablement 232
 - WINS 100, 105
- WINS 100, 105, 177, 180, 197, 201, 211
- workload 95
- Workspace On-Demand 28, 66, 174, 320, 324

Y

- Yahoo 244

Z

- zone 95, 101, 312
 - primary 103
- zone key 127
- zone of authority 91, 92
- zone transfer 95, 96

ITSO Redbook Evaluation

Beyond DHCP — Work Your TCP/IP Internetwork with Dynamic IP
SG24-5280-00

Your feedback is very important to help us maintain the quality of ITSO redbooks. **Please complete this questionnaire and return it using one of the following methods:**

- Use the online evaluation form found at <http://www.redbooks.ibm.com>
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to redbook@us.ibm.com

Which of the following best describes you?

☐ **Customer** ☐ **Business Partner** ☐ **Independent Software Vendor** ☐ **IBM employee**
☐ **None of the above**

Please rate your overall satisfaction with this book using the scale:
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction _____

Please answer the following questions:

Was this redbook published in time for your needs? Yes____ No____

If no, please explain:

What other redbooks would you like to see published?

Comments/Suggestions: (THANK YOU FOR YOUR FEEDBACK!)

