Troubleshooting DHCP Problems in Cable Networks using Cisco Network Registrar Debugs

Contents

Introduction

The Data-over-Cable Service Interface Specifications (DOCSIS) mandate that cable modems negotiate their IP addresses with DHCP. Cisco Network Registrar (CNR) provides comprehensive Domain Name System (DNS) and DHCP administrative functionality. CNR also provides TFTP server functionality.

DHCP negotiation is a common problem in cable network environments that transport IP data. You can enable debugs on CNR to troubleshoot DHCP negotiation. This document begins with an explanation of CNR functionality, then explains how to enable the debugs. Finally, it provides common examples of situations where cable modems do not come online or where customer premises equipment (CPE) behind the cable modems can not connect to the Internet.

Prerequisites

Requirements

This document applies to CNR 5.0.

Components Used

The information in this document is based on these software and hardware versions:

- UNIX systems
  - Solaris
  - HP/UX
  - AIX
- Windows NT
- Windows 2000

Note: The GUI interface for UNIX systems is only available in Solaris.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Conventions

For more information on document conventions, refer to the Cisco Technical Tips Conventions.

Directory Structure in CNR

UNIX Systems

The directory structure for UNIX starts at this directory:

```
/opt/nwreg2
```

The directory contains these subdirectories:

```
skyshark# ls -l

total 18
drwxr-xr-x   2 root     bin         1024 Mar 28 18:34 bin/
drwxr-xr-x   2 root     bin          512 Mar 28 18:35 conf/
drwxr-xr-x   3 root     bin          512 Mar 28 18:33 docs/
drwxr-xr-x   3 root     bin          512 Mar 28 18:31 examples/
drwxr-xr-x   3 root     bin          512 Mar 28 18:31 extensions/
drwxr-xr-x   3 root     bin         1024 Mar 28 18:35 lib/
drwxr-xr-x   2 root     bin          512 Mar 28 18:33 misc/
drwxr-xr-x   2 root     bin          512 Apr  2 18:39 usrbin/
drwxr-xr-x   5 root     bin          512 Mar 28 18:31 WebUI/
```

The subdirectories contain these components:

- `bin`—Executable programs.

- `conf`—Configuration files.
- `lib`—Libraries used by executable files.
- `usrbin`—The subdirectory in which you launch the GUI or nrcmd (Network Registrar Command, the CNR command line interface [CLI]).

To launch the GUI, issue **ntwkreg**. To launch the CLI, issue **nrcmd**.

The /opt/nwreg2/usrbin directory contains these files:

```
skyshark# pwd

/opt/nwreg2/usrbin

skyshark# ls -l

total 11422
-r-xr-xr-x   1 root     bin          980 Mar 28 18:35 aicstatus*
-r-xr-xr-x   1 root     bin          365 Mar 28 18:34 cnrFailoverConfig*
-r-xr-xr-x   1 root     bin          179 Mar 28 18:34 mcdadmin*
-r-xr-xr-x   1 root     bin          180 Mar 28 18:35 mcdshadow*
-r-xr-xr-x   1 root     bin          385 Mar 28 18:34 nrcmd*
-r-xr-xr-x   1 root     bin         1986 Mar 28 18:35 ntwkreg*
```

The databases and the logs are in the /var/nwreg2 directory. These files should have write access.

```
/var/nwreg2

skyshark# ls -l

total 6
drwxr-xr-x   9 root     other        512 Mar 28 18:36 data/
drwxrwxrwt   3 root     other        512 APR 16 09:07 logs/
drwxr-xr-x   2 root     other        512 Mar 28 18:42 temp/
```

The subdirectories contain these components:

- `data`—Location of the database data and backup files:
  - `db`—The active database.
  - `db.bak`—A copy of the database. This copy is made every night at 11:45 PM (server time).
  - `dns`—The cache file and current authoritative zone field that is being read by the server and passed to secondaries in a zone transfer.
- `logs`—This directory contains the log files. A common mistake is to look in the /opt/ subdirectory. The easiest way to remember is that the logs are written by a server and, therefore, must be in a directory with write access. The logs are often used to troubleshoot, and they are what you use the most in this document.
- `temp`—Locked temporary files that are used to run AIC Server Agent.

On UNIX, there are several processes related to running CNR. To check the status, issue this command:

```
skyshark# /opt/nwreg2/usrbin/aicstatus

Server Agent running     (pid: 112)
```

```
MCD lock manager running (pid: 118)
MCD server running       (pid: 116)
DNS server running       (pid: 119)
DHCP server running      (pid: 120)
```

Run only one instance of each Server Agent.

To stop and start the process, issue these commands:

```
skyshark# /etc/init.d/aicservagt stop

skyshark# /etc/init.d/aicservagt start

# Starting AIC Server Agent for Network Registrar
```

Windows Systems

For Windows NT and Windows 2000, the structure is similar. If you installed CNR in the C: drive, this is the installation directory:

```
C:\Program Files\Network Registrar\
```

That directory contains these files and subdirectories:

```
C:\Program Files\Network Registrar> dir

 Volume in drive C is W2K
 Volume Serial Number is D439-C697

 Directory of C:\Program Files\Network Registrar

01/24/2001  03:22p      <DIR>          .
01/24/2001  03:22p      <DIR>          ..
01/24/2001  03:22p      <DIR>          BIN
04/14/2001  11:46p      <DIR>          DATA
01/24/2001  03:23p              15,037 DeIsL1.isu
01/24/2001  03:22p      <DIR>          DOCS
01/24/2001  03:22p      <DIR>          EXAMPLES
01/24/2001  03:21p      <DIR>          EXTENSIONS
01/24/2001  03:22p      <DIR>          lib
04/09/2001  08:38a      <DIR>          LOGS
01/24/2001  03:22p      <DIR>          MISC
12/25/2000  05:12p               2,083 README.TXT
01/24/2001  03:21p      <DIR>          TEMP
12/25/2000  10:12p              58,880 unregistrar.dll
01/24/2001  03:21p      <DIR>          WebUI
               3 File(s)         76,000 bytes
              12 Dir(s)   1,426,918,400 bytes free
```

The directory structure in NT is different than in Unix. NT is more flexible because all files are located in one directory and the specific read-only files are flagged.

On Windows, there is only one process that is running: AIC Server Agent 2.0. In Window NT, choose **Start > Settings > Control Panel > Services** to control it.

Server Status Monitor

You can monitor the status of the DNS, DHCP, and TFTP servers in CNR with the Server Status Monitor. This monitor displays the aspects of the health of the specified server.

To add servers to the Server Status Monitor, use this procedure to drag and drop servers into the Status Monitor:

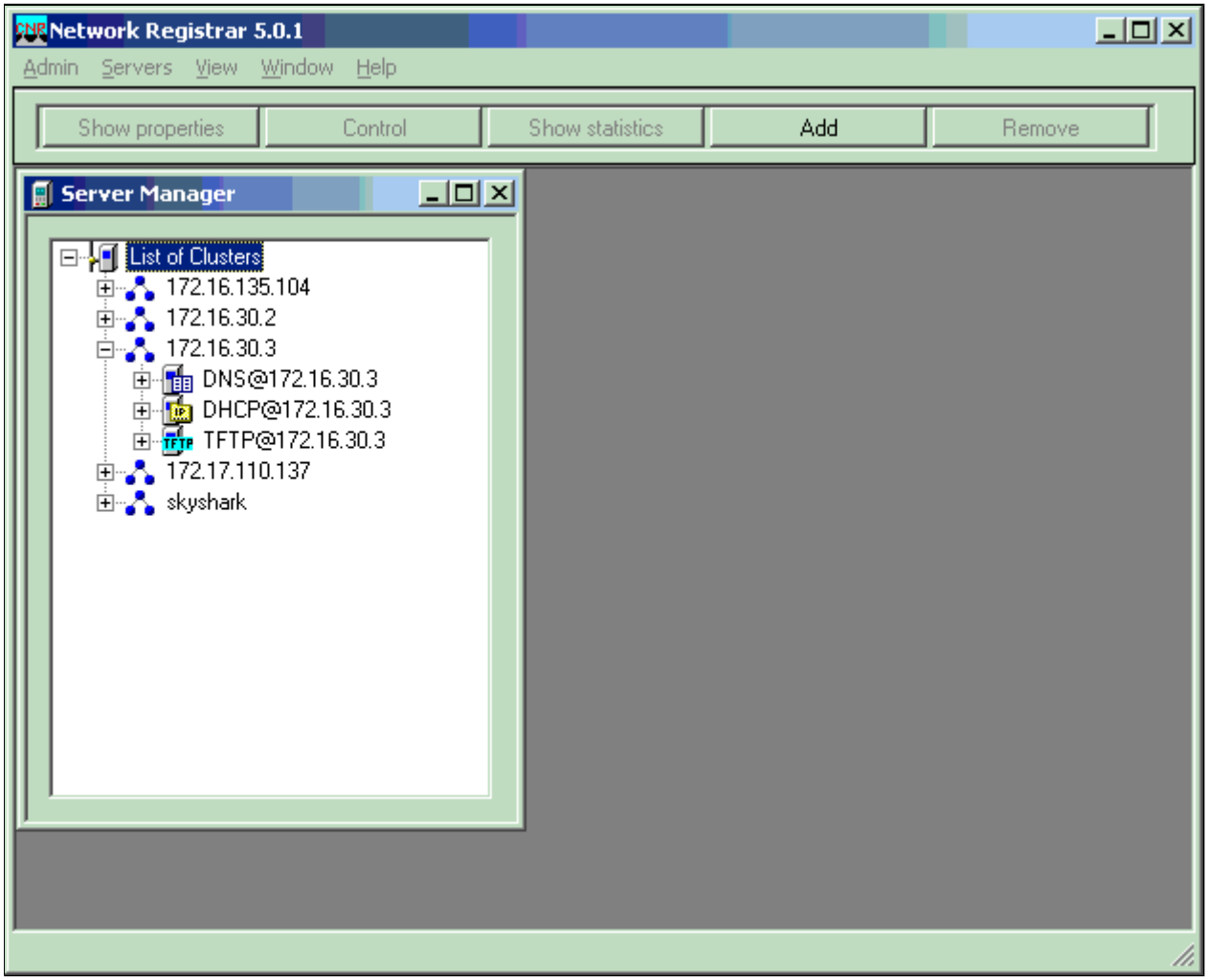
1. Launch the CNR GUI:
    a. In the UNIX operating system, launch Xterm and follow this procedure:
        a. Issue the **xhost +** command.
        b. Telnet to the UNIX system that hosts the CNR server.
        c. Issue these commands:
            - **setenv TERM xterm**
            - **setenv DISPLAY *your-local-ip-address*:0.0**
        d. Issue this command to launch the GUI:
            - **/opt/nwreg2/usrbin/ntwkreg &**

              Note: The **&** allows you to use that window for other commands.
    b. In the Windows operating system, choose **Start > Programs > Network Registrar**.
2. Once you launch the GUI, the system asks you for the username and passwords. When CNR is first installed, it uses the username **admin** and the password **changeme**.

   Caution: Change this password.
3. Click the + sign next to the cluster that you want to monitor.

   You now see a screen similar to Figure 1.

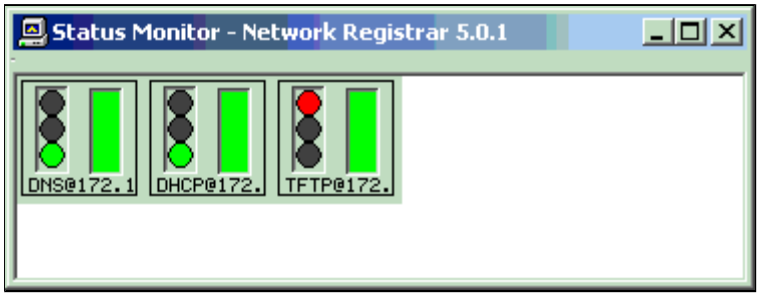
   Figure 1 – Server Manager Window for CNR 5.0.1



4. Right-click the server that you want to monitor and choose **Add to Status Monitor**. Do this for each server that you want to monitor.

   The Status Monitor window shows a green light next to the servers that are running. Figure 2 shows that the DNS and DHCP servers for cluster 172.16.30.3 are active, while the TFTP server for the same cluster is not (it shows a red light).

   Figure 2 – Status Monitor Window for CNR 5.0.1



   Note: If you want to remove a server from the Status Monitor window, right-click the server and choose **Remove**.

DHCP Debugs Settings

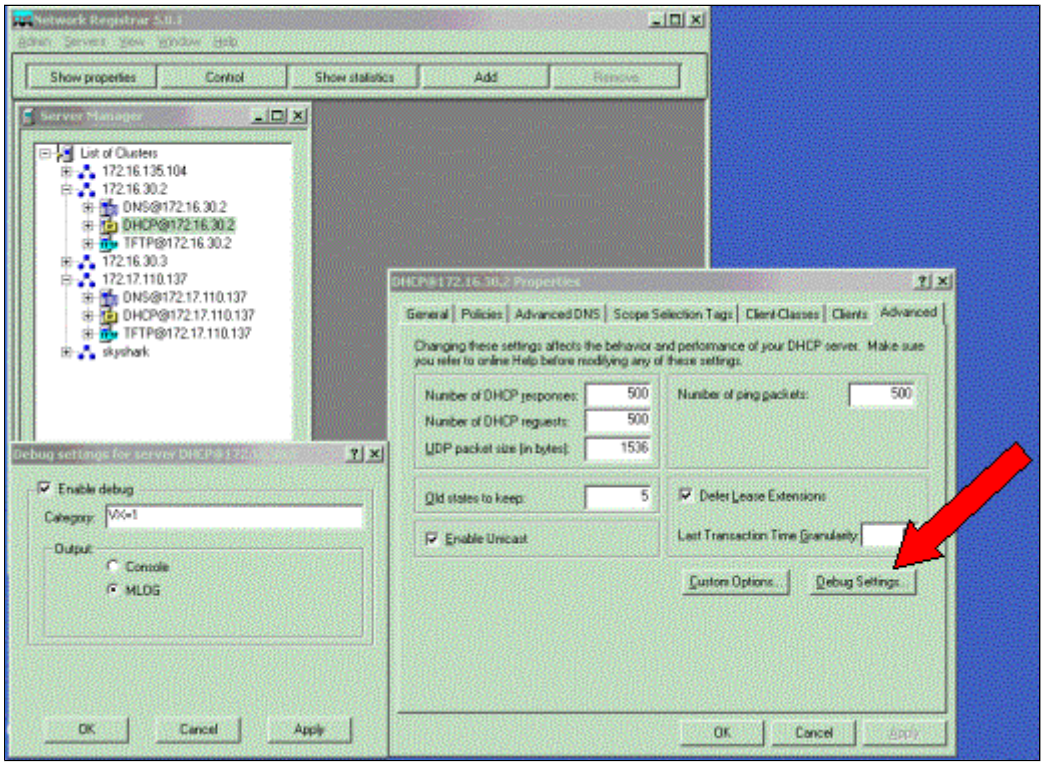
You to have enough information in the debugs settings to troubleshoot a DHCP problem. The information is saved in the log files. You can use both the GUI and the CLI (nrcmd) to set the debugs in CNR.

Using the GUI to Set the Debugs for DHCP

To use the GUI to set the debugs for DHCP, use this procedure:

1. From the Server Manager, select the server for which you want to set the debug option.
2. Click **Show properties**.
3. Click the **Advanced** tab on the Properties dialog box.
4. Click **Debug Settings** (indicated by the red arrow in Figure 3).

Figure 3 – Dialog to Set the Debug Levels for DHCP



5. In the Debug settings dialog box, check **Enable debug**.
6. In the Category field, enter one of the setting from Table 1.

Table 1 – Settings, Levels, and Description for DHCP Debug Settings

Server Category (DHCP)	Level	Description
VX=	1	Incoming and outgoing detailed packet trace.
KP=	1-9	Dynamic DNS packet trace and full details on all the messages to and from Lightweight Directory Access Protocol (LDAP), including value attributes.
Q=	1-9	Class of Service (CoS) trace.

Y=	1	The log-failover-detail log setting.
Y=	2	Moderate detail on failover.
Y=	3	Formatted failover packets. Does not include poll packets. This gives the same output as VX=1, but only for failover packets.
Y=	4	Formatted failover packets, polls included.
A-LZ-Z=	9	All DHCP logging.

7. Click the **MLOG** radio button, which sends the output to the appropriate log file.
8. Click **OK** in the Debug settings dialog box and then in the Properties dialog box.

Using the CLI to Set the Debugs for DHCP

You can also set debugs settings with the CLI (nrcmd).

This is the format of the command that you issue:

```
nrcmd> server server-type setDebug categories=level
```

- *server-type*—The server in question; in the example for this section, it is **dhcp**.
- *categories*—Correspond to the **Server Category (DHCP)** column of Table 1.
- *level*—One of the numeric values, from the **Level** column of Table 1, that corresponds to the *categories*.

If you do not specify all of the required arguments (variables), then you will see this message:

```
nrcmd> dhcp setDebug

310 Too few arguments - usage: server <server> setDebug <categories>=<level>..
```

To deactivate all debug settings, issue the **unsetDebug** command:

```
nrcmd> server dhcp unsetDebug

100 OK
```

Example 1

To set the debug settings for the DHCP server to incoming and outgoing detailed packet trace (VX=1), issue this command:

```
nrcmd> server dhcp setDebug VX=1

100 OK
```

Note: The `100 OK` message indicates that the command is accepted. If you make a mistake, you see a message similar to this one:

```
nrcmd> server dhcp setDebug= vx=1

306 Unknown command - dhcp method 'setDebugs='
```

In that error, an extra, incorrect = sign was entered after **setDebug**.

Example 2

To set the debug setting for the DHCP server for the most detailed DHCP logging, issue this command:

```
nrcmd> server dhcp setDebug A-LN-Z=9

100 OK
```

Sometimes the CLI accepts higher level numbers than those that are specified in Table 1, Table 2, or Table 3. Such higher debug levels do not provide more detailed information than those specified in these tables.

In this example, the level is set to 100, and the command is accepted; but the details that are going to be sent to the logs are the same as those for level 9:

```
nrcmd> dhcp setDebug A-LN-Z=100

100 OK
```

DNS Debug Settings

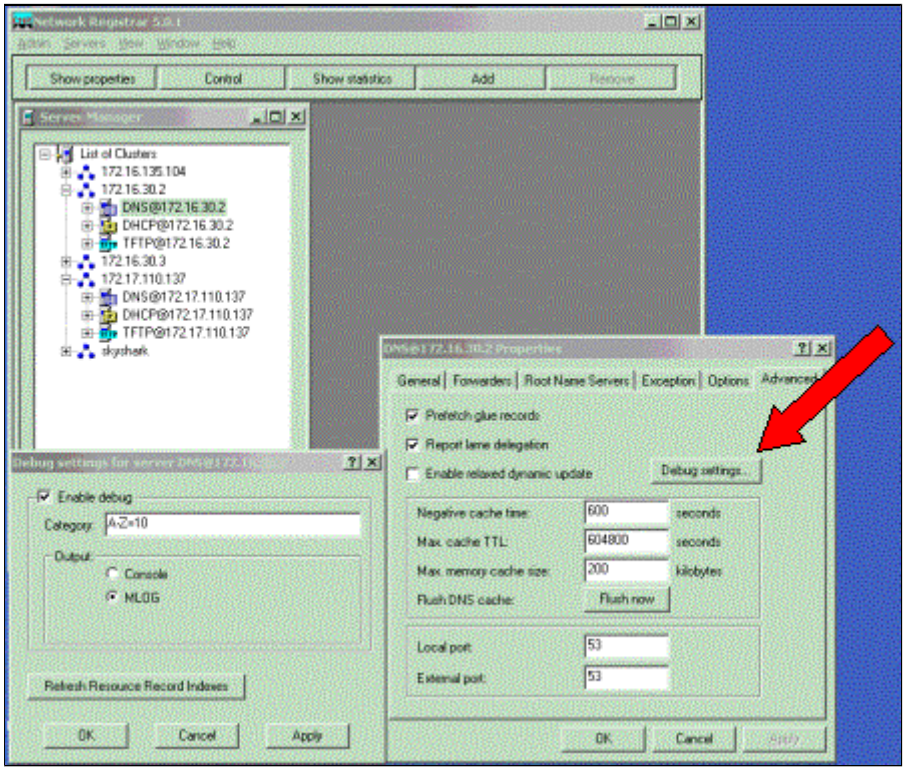
You can use DNS debug settings to troubleshoot DNS problems. You can use both the GUI and the CLI to set the debug levels for DNS.

Using GUI to Set the Debugs for DNS

To use the GUI to set the debugs for DNS, use this procedure:

1. From the Server Manager, select the server for which you want to set the debug option.
2. Click **Show properties**.
3. Click the **Advanced** tab on the Properties dialog box.
4. Click **Debug Settings** (indicated by the red arrow in Figure 4).

   Figure 4 – Dialog to Set the Debug Levels for DHCP



5. In the Debug settings dialog box, check **Enable debug**.
6. In the Category field, enter one of the setting from Table 2.

Table 2 – Settings, Levels, and Description for DHCP Debug Settings

Server Category (DNS)	Level	Description
D		Basic DNS tracing
	1	Failures, errors, configuration oddities, some configuration details.
	2	Less significant errors. Format errors in responses, forwards, re-forwards.
	3	Recoverable failures, illegal name server, configuration slave mode or transfer (XFER) client major state changes.
	4	Incremental transfer (IXFR) server packing each record, IXFR or transfer zone contents (AXFR) and Zone Transfers.
	5	Per-name completion of background zone loading.
	6	Zone History trimming schedule and completion. Per-packet XFER client logging.
U		Dynamic Update tracing

	1	Errors, failures, non-success replies.
	2	Per-packet logging source, zone, ld, prereq, and update resource record counts.
	3	Per-packet inbound packet logging, packet or request validation errors.
	4	Per-packet outbound packet basic logging.
	5	Inbound duplicates on zone, new inbound requests on zone, XFER client reaction to notify requests on zone.
P		Packets
	1	Per-query packet after basic packet validation.
	2	Per-inbound packet, pre-validation logging.
	3	Per-outbound packet, response data logging.
DNUP		All DNS logging
A-Z	10	*All* inbound and outbound packets, requests, forwarded messages, dynamic updates, notify messages, incremental and full zone transfers, and many verbose, function-specific decision tree information messages for all internal DNS server, database, and library sub-systems.

7. Click the **MLOG** radio button, which sends the output to the appropriate log file.
8. Click **OK** in the Debug settings dialog box and then in the Properties dialog box.

Using CLI to Set the Debugs For DNS

You can set the debugs settings with the CLI (nrcmd).

This is the format of the command that you issue:

```
nrcmd> server server-type setDebug categories=level
```

- *server-type*—The server in question; in this example for the section, it is **dns**.
- *categories*—Correspond to the **Server Category (DNS)** column of Table 2.
- *level*—One of the numeric values, from the **Level** column of Table 2, that corresponds to the *categories*.

To deactivate all debug settings, issue the **unsetDebug** command. It deactivates all settings.

```
nrcmd> server dns unsetDebug

100 OK
```

Example

```
nrcmd> server dns setDebug D=5

100 OK

nrcmd> server dns setDebug AZ=10

100 OK
```

TFTP Debug Settings

When you are having problems with the TFTP server, use the CLI to set the debug levels. It is not possible to use the GUI for this purpose. Table 3 shows the debugs levels that can be set in CNR for the TFTP server.

Table 3 – Settings, Levels, and Description for TFTP Server Debug Settings

Server Category (TFTP)	Level	Description
C	1-3	Server configuration.
D	1-3	Statistics.
E	1-3	CSCR extension object.
F	1-3	File handling.
P	1-3	Packet handling.
S	1-3	TFTP session handling.
T	1-3	Timer handling.

The **Levels** above correspond to these debug types:

- 1—Unexpected conditions.
- 2—More detailed information.
- 3—Every possible debug.

Example

To set the debug setting for the TFTP CNR server for the most detailed server configuration logging, issue this command:

```
nrcmd> server tftp setDebug C=3

100 OK
```

Note: If you do not see the `100 OK` message, then CNR did not accept the command.

To deactivate all debug settings, issue the **unsetDebug** command:

```
nrcmd> server tftp unsetDebug
```

```
100 OK
```

Configuration Issues: The Client Does Not Get an IP Address

One of the most common problems with the use of DHCP server of the CNR in cable environments is that clients—cable modems and the CPE behind them—do not receive an IP address. If this is the case, cable modems get stuck in the init(d) state. For details on this situation, refer to Troubleshooting uBR Cable Modems Not Coming Online. There are several possible causes of this problem. The rest of this document discusses each of the reasons.

Is the DHCP DISCOVER Packet Reaching CNR?

CNR logs, even at the default level, show enough information to determine if CNR received the packet. You can check if the MAC address of the client Client Identifier (CID) appears in a DHCPDISCOVER packet. The left-most byte of the CID indicates the htype=1 for Ethernet, so the real MAC address is the right-most six bytes. With the default logging level, you see this as a portion of the name_dhcp_1_log (in C:\Program Files\Network Registrar\LOGS in Windows NT):

```
!--- Output suppressed.

08/24/2000 17:40:09 name/dhcp/1 Activity Server 0 04619 Server received
                    a DHCPDISCOVER packet 'R1' from:
                    Host: 'dell-port-pc' CID: 01:00:10:a4:ff:61:8e
                    with IP source address: 0.0.0.0 via: Interface 10.200.68.200,
                    1 in use.
```

This output shows that the DHCPDISCOVER packet was received.

This is the same output with a more detailed logging; VX=1 is the debug level that is set:

```
08/24/2000 17:45:19 name/dhcp/1 Info Protocol 0 04935 R1:
                    -----  RECEIVED -- R1 -----
08/24/2000 17:45:19 name/dhcp/1 Info Protocol 0 04935 R1:
                    ->  port = 68 received from  = 0.0.0.0
08/24/2000 17:45:19 name/dhcp/1 Info Protocol 0 04935 R1:
                    ->  packet length = 300
08/24/2000 17:45:19 name/dhcp/1 Info Protocol 0 04935 R1:
                    ->  op = 1 request
08/24/2000 17:45:19 name/dhcp/1 Info Protocol 0 04935 R1:
                    ->  htype = 1 ethernet      hlen = 6
08/24/2000 17:45:19 name/dhcp/1 Info Protocol 0 04935 R1:
                    ->  hops =   0
08/24/2000 17:45:19 name/dhcp/1 Info Protocol 0 04935 R1:
                    ->  xid = 0xec9e secs = 0 flags = 0x0
08/24/2000 17:45:19 name/dhcp/1 Info Protocol 0 04935 R1:
                    ->  ciaddr = 0.0.0.0
08/24/2000 17:45:19 name/dhcp/1 Info Protocol 0 04935 R1:
                    ->  yiaddr = 0.0.0.0
08/24/2000 17:45:19 name/dhcp/1 Info Protocol 0 04935 R1:
                    ->  siaddr = 0.0.0.0
08/24/2000 17:45:19 name/dhcp/1 Info Protocol 0 04935 R1:
                    ->  giaddr = 0.0.0.0
08/24/2000 17:45:19 name/dhcp/1 Info Protocol 0 04935 R1:
                    ->  chaddr = 0:10:a4:ff:61:8e
08/24/2000 17:45:19 name/dhcp/1 Info Protocol 0 04935 R1:
                    ->     dhcp-message-type = 1 discover
08/24/2000 17:45:19 name/dhcp/1 Info Protocol 0 04935 R1:
                    ->     dhcp-client-identifier =1 0 16 164 255 97 142
08/24/2000 17:45:19 name/dhcp/1 Info Protocol 0 04935 R1:
                    ->     ethernet? = 0:10:a4:ff:61:8e
```

```
08/24/2000 17:45:19 name/dhcp/1 Info Protocol 0 04935 R1:
                    ->     dhcp-requested-address = 10.200.68.100
08/24/2000 17:45:19 name/dhcp/1 Info Protocol 0 04935 R1:
                    ->     host-name = "dell-port-PC"
08/24/2000 17:45:19 name/dhcp/1 Info Protocol 0 04935 R1:
                    ->     end
08/24/2000 17:45:19 name/dhcp/1 Info Protocol 0 04935 R1:
                    ->  sname = ""
08/24/2000 17:45:19 name/dhcp/1 Info Protocol 0 04935 R1:
                    ->  file  = ""
08/24/2000 17:45:19 name/dhcp/1 Info Protocol 0 04935 R1:
                    ----- END OF RECEIVED -- R1 -----
```

As you can see in this output, you get the same information about the DHCPDISCOVER; but, with the debug level set to VX=1, more detailed information about the packet itself is provided.

Does CNR Have a Suitable Scope for the Client?

DHCP does not understand the subnet mask concept of IP. When CNR receives a DHCPDISCOVER, it looks at the GIADDR field or hops field: if they are empty or equal to 0, then CNR assumes that the request comes from the local network. It looks at the IP address of the interface on which it received the packet, and it uses that one to select a scope. If the GIADDR field is not empty, then the DHCPDISCOVER was forwarded by a DHCP relay agent. The DHCP relay agent is usually configured on Cisco routers with the **ip helper-address** command. In this case, CNR uses the IP address in the GIADDR field to select a scope. That IP address was of the router interface that received the broadcast request of the client. In both cases, because there is no subnet mask information in the request, CNR does a best match in all configured scopes to select a good one.

Suppose in CNR that you configure 10.0.0.0/8: this will be good for a request coming from 10.200.68.200. If there is a more specific one, like 10.200.0.0/16 or 10.200.68.0/24, then the one with the longest mask is chosen. It is better to create scopes with the same subnet mask of the network where they are assigned. This is because the scope subnet mask is the subnet mask that is assigned to the clients and, on a network segment, all of the hosts should share the same subnet mask. It is possible to assign to the clients a subnet mask that is different from the one that is defined in the scope with the DHCP option **get-subnet-mask-from-policy**. In this example, CNR does not have a scope, so it discards the request:

```
08/24/2000 17:45:19 name/dhcp/1 Warning Protocol 0 04663
                    Received DHCPDISCOVER packet but found no Scopes
                    for source network = '10.200.68.200'. Dropping packet.
```

The next sample output shows a packet that is forwarded by a relay agent. Notice the hops and GIADDR field values, and compare them to the values in the previous sample output. The result is the same because CNR does not have a scope that suits 10.200.71.1.

```
08/24/2000 18:11:23 name/dhcp/1 Info Protocol 0 04935 R61:
                    ----- RECEIVED -- R61 -----
08/24/2000 18:11:23 name/dhcp/1 Info Protocol 0 04935 R61:
                    ->  port = 67    received from = 10.200.71.1
08/24/2000 18:11:23 name/dhcp/1 Info Protocol 0 04935 R61:
                    ->  packet length = 296
08/24/2000 18:11:23 name/dhcp/1 Info Protocol 0 04935 R61:
                    ->  op =  1 request
08/24/2000 18:11:23 name/dhcp/1 Info Protocol 0 04935 R61:
                    ->  htype = 1  ethernet    hlen = 6
08/24/2000 18:11:23 name/dhcp/1 Info Protocol 0 04935 R61:
                    ->  hops = 1
08/24/2000 18:11:23 name/dhcp/1 Info Protocol 0 04935 R61:
                    ->  xid = 0x2127  secs = 0  flags = 0x8000  broadcast
08/24/2000 18:11:23 name/dhcp/1 Info Protocol 0 04935 R61:
                    ->  ciaddr = 0.0.0.0
08/24/2000 18:11:23 name/dhcp/1 Info Protocol 0 04935 R61:
```

```
                    ->  yiaddr = 0.0.0.0
08/24/2000 18:11:23 name/dhcp/1 Info Protocol 0 04935 R61:
                    ->  siaddr = 0.0.0.0
08/24/2000 18:11:23 name/dhcp/1 Info Protocol 0 04935 R61:
                    ->  giaddr = 10.200.71.1
08/24/2000 18:11:23 name/dhcp/1 Info Protocol 0 04935 R61:
                    ->  chaddr = 0:1:96:59:47:c1
08/24/2000 18:11:23 name/dhcp/1 Info Protocol 0 04935 R61:
                    ->    dhcp-message-type = 1 discover
08/24/2000 18:11:23 name/dhcp/1 Info Protocol 0 04935 R61:
                    ->    dhcp-max-message-size = 1152
08/24/2000 18:11:23 name/dhcp/1 Info Protocol 0 04935 R61:
                    ->    dhcp-client-identifier = 1 0 1 150 89 71 193
08/24/2000 18:11:23 name/dhcp/1 Info Protocol 0 04935 R61:
                    ->    ethernet? = 0:1:96:59:47:c1
08/24/2000 18:11:23 name/dhcp/1 Info Protocol 0 04935 R61:
                    ->    dhcp-parameter-request-list = 20
08/24/2000 18:11:23 name/dhcp/1 Info Protocol 0 04935 R61:
                    ->        1    subnet-mask
08/24/2000 18:11:23 name/dhcp/1 Info Protocol 0 04935 R61:
                    ->        2    time-offset
08/24/2000 18:11:23 name/dhcp/1 Info Protocol 0 04935 R61:
                    ->        4    time-servers
08/24/2000 18:11:23 name/dhcp/1 Info Protocol 0 04935 R61:
                    ->        66   tftp-server
08/24/2000 18:11:23 name/dhcp/1 Info Protocol 0 04935 R61:
                    ->        128  mcns-sec-server
08/24/2000 18:11:23 name/dhcp/1 Info Protocol 0 04935 R61:
                    ->        3    routers
08/24/2000 18:11:23 name/dhcp/1 Info Protocol 0 04935 R61:
                    ->        7    log-servers
08/24/2000 18:11:23 name/dhcp/1 Info Protocol 0 04935 R61:
                    ->        67   boot-file
08/24/2000 18:11:23 name/dhcp/1 Info Protocol 0 04935 R61:
                    ->    dhcp-class-identifier = "docsis1.0"
08/24/2000 18:11:23 name/dhcp/1 Info Protocol 0 04935 R61:
                    ->    dhcp-option-overload = 3
08/24/2000 18:11:23 name/dhcp/1 Info Protocol 0 04935 R61:
                    ->    relay-agent-info = 1 4 128 6 0 9 2 6 0 1 150 89 71 193
08/24/2000 18:11:23 name/dhcp/1 Info Protocol 0 04935 R61:
                    ->
08/24/2000 18:11:23 name/dhcp/1 Info Protocol 0 04935 R61:
                    ->
08/24/2000 18:11:23 name/dhcp/1 Info Protocol 0 04935 R61:
                     ----- END OF RECEIVED -- R61 -----
08/24/2000 18:11:23 name/dhcp/1 Warning Protocol 0 04663
                    Received DHCPDISCOVER packet but found no Scopes
                    for source network = '10.200.71.1'. Dropping packet.
```

Are There Free Addresses In the Scope?

It is possible that the scopes are configured in such a way that the pool of addresses is too small. In such cases, you can commonly run out of addresses in the pool. This sample output shows the debug messages when a client is attempting to get an IP address but the scope already uses all of its addresses in the pool:

```
08/24/2000 19:14:26 name/dhcp/1 Warning Server 0 04440
                    No more leases are AVAILABLE, unable to respond
                    to DHCP DISCOVER Request: R9 = from Client: Host:
                    dell-port-PC CID: 01:00:10:a4:ff:61:8e in Network:
                    10.200.68.0-255.255.255.0 via: Interface 10.200.68.200
```

Does the Offer Reach the Client?

- Once CNR selects a scope and creates an offer packet, the information must be transmitted to the

client.

Note: When the packet is destined to a relay agent, ensure that the CNR machine has a route to the GIADDR.

- In the case of these examples, ensure that you can ping from the CNR machine 10.200.71.1.
- Remember that, after a lease is offered, there are a few other steps before that becomes a real IP address on the network.
- You must see, from the same client, one or more DHCPREQUEST and one or more DHCPOFFER from the server to the client. These are used to request and obtain DHCP options.
- You must see a final DHCPACK from the server to the client, which terminates the DHCP process.
- If you only see DHCPOFFER but nothing else, ensure that the packets reach the client and that the subsequent DHCPREQUEST comes back to the server.

This is a log entry that shows that a lease has been granted (this information provides the client's MAC address, the IP address that is assigned, and the lease expiration date):

```
08/24/2000 13:13:15 name/dhcp/1 Activity Protocol 0 04994 10.200.68.200
                    Lease granted to Host: dell-port-PC CID: 01:00:10:a4:ff:61:8e
                    packet 'R207' until Thu, 31 Aug 2000 13:13:15  +0200. 320 ms.
```

Is There Any Other Server on the Same Network With the Same Scope Configured?

In such a case, both servers receive a DHCPDISCOVER and make their offer, but the client only chooses one. It indicates, in the **dhcp-server-identifier**, from which server it accepts an offer. If the other server sees the DHCPREQUEST for the same IP address that it has offered—but the **dhcp-server-identifier** points to another server—then it deactivates the lease, to prevent possible duplicate addresses.

The last line in this sample output shows this deactivation:

```
09/01/2000 12:21:05 name/dhcp/1 Info Protocol 0 04935 R7:
                    ----- RECEIVED -- R7 -----
09/01/2000 12:21:05 name/dhcp/1 Info Protocol 0 04935 R7:
                    ->   port = 68 received from = 0.0.0.0
09/01/2000 12:21:05 name/dhcp/1 Info Protocol 0 04935 R7:
                    ->   packet length = 300
09/01/2000 12:21:05 name/dhcp/1 Info Protocol 0 04935 R7:
                    ->   op = 1 request
09/01/2000 12:21:05 name/dhcp/1 Info Protocol 0 04935 R7:
                    ->   htype = 1 ethernet hlen =3D 6
09/01/2000 12:21:05 name/dhcp/1 Info Protocol 0 04935 R7:
                    ->   hops =  0
09/01/2000 12:21:05 name/dhcp/1 Info Protocol 0 04935 R7:
                    ->   xid =0xddaadeaa   secs = 0 flags = 0x0
09/01/2000 12:21:05 name/dhcp/1 Info Protocol 0 04935 R7:
                    ->   ciaddr = 0.0.0.0
09/01/2000 12:21:05 name/dhcp/1 Info Protocol 0 04935 R7:
                    ->   yiaddr = 0.0.0.0
09/01/2000 12:21:05 name/dhcp/1 Info Protocol 0 04935 R7:
                    ->   siaddr = 0.0.0.0
09/01/2000 12:21:05 name/dhcp/1 Info Protocol 0 04935 R7:
                    ->   giaddr = 0.0.0.0
09/01/2000 12:21:05 name/dhcp/1 Info Protocol 0 04935 R7:
                    ->   chaddr = 0:10:a4:ff:61:8e
09/01/2000 12:21:05 name/dhcp/1 Info Protocol 0 04935 R7:
                    ->     dhcp-message-type = 3 request
```

```
09/01/2000 12:21:05 name/dhcp/1 Info Protocol 0 04935 R7:
                    ->     dhcp-client-identifier = 1 0 16 164 255 97 142
09/01/2000 12:21:05 name/dhcp/1 Info Protocol 0 04935 R7:
                    ->     ethernet? = 0:10:a4:ff:61:8e
09/01/2000 12:21:05 name/dhcp/1 Info Protocol 0 04935 R7:
                    ->     dhcp-requested-address = 10.200.68.201
09/01/2000 12:21:05 name/dhcp/1 Info Protocol 0 04935 R7:
                    ->     dhcp-server-identifier = 10.200.68.17
09/01/2000 12:21:05 name/dhcp/1 Info Protocol 0 04935 R7:
                    ->     hostname = "dell-port-PC"
09/01/2000 12:21:05 name/dhcp/1 Info Protocol 0 04935 R7:
                    ->     dhcp-parameter-request-list = 20
09/01/2000 12:21:05 name/dhcp/1 Info Protocol 0 04935 R7:
                    ->         1   subnet-mask
09/01/2000 12:21:05 name/dhcp/1 Info Protocol 0 04935 R7:
                    ->         3   routers
09/01/2000 12:21:05 name/dhcp/1 Info Protocol 0 04935 R7:
                    ->         15  domain-name
09/01/2000 12:21:05 name/dhcp/1 Info Protocol 0 04935 R7:
                    ->         6   domain-name-servers
09/01/2000 12:21:05 name/dhcp/1 Info Protocol 0 04935 R7:
                    ->         44  netbios-name-servers
09/01/2000 12:21:05 name/dhcp/1 Info Protocol 0 04935 R7:
                    ->         46  netbios-node-type
09/01/2000 12:21:05 name/dhcp/1 Info Protocol 0 04935 R7:
                    ->         47  netbios-scope
09/01/2000 12:21:05 name/dhcp/1 Info Protocol 0 04935 R7:
                    ->   vendor-encapsulated-options = 55 2 0 0
09/01/2000 12:21:05 name/dhcp/1 Info Protocol 0 04935 R7:
                    ->   end
09/01/2000 12:21:05 name/dhcp/1 Info Protocol 0 04935 R7:
                    ->   sname = ""
09/01/2000 12:21:05 name/dhcp/1 Info Protocol 0 04935 R7:
                    ->   file = ""
09/01/2000 12:21:05 name/dhcp/1 Info Protocol 0 04935 R7:
                    ----- END OF RECEIVED -- R7 -----
09/01/2000 12:21:05 name/dhcp/1 Activity Protocol 0 04993 10.200.68.201
                    Lease offered to Host: dell-port-PC CID: 01:00:10:a4:ff:61:8e
                    packet 'R5' until Fri, 01 Sep 2000 12:23:05 +0200. 150 ms.
09/01/2000 12:21:05 name/dhcp/1 Error Protocol 0  04684 Client:
                    'Host: dell-port-PC CID: 01:00:10:a4:ff:61:8e ' sent a
                    REQUEST for Lease: '10.200.68.201' to Server: '10.200.68.17'
                    instead of us. Marking Lease UNAVAILABLE
```

If the server sees a DHCPREQUEST for a different IP address, it simply logs it. In this sample output, the other server (10.200.68.17) offered 10.200.68.201:

```
09/01/2000 12:19:33 name/dhcp/1 Info Protocol 0 04935 R3:
                    ----- RECEIVED -- R3 -----
09/01/2000 12:19:33 name/dhcp/1 Info Protocol 0 04935 R3:
                    ->  port = 68    received from = 0.0.0.0
09/01/2000 12:19:33 name/dhcp/1 Info Protocol 0 04935 R3:
                    ->  packet length = 300
09/01/2000 12:19:33 name/dhcp/1 Info Protocol 0 04935 R3:
                    ->  op = 1 request
09/01/2000 12:19:33 name/dhcp/1 Info Protocol 0 04935 R3:
                    ->  htype = 1  ethernet    hlen = 6
09/01/2000 12:19:33 name/dhcp/1 Info Protocol 0 04935 R3:
                    ->  hops = 0
09/01/2000 12:19:33 name/dhcp/1 Info Protocol 0 04935 R3:
                    ->  xid = 0x8a7d8b7d  secs = 0  flags = 0x0
09/01/2000 12:19:33 name/dhcp/1 Info Protocol 0 04935 R3:
                    ->  ciaddr = 0.0.0.0
09/01/2000 12:19:33 name/dhcp/1 Info Protocol 0 04935 R3:
                    ->  yiaddr = 0.0.0.0
```

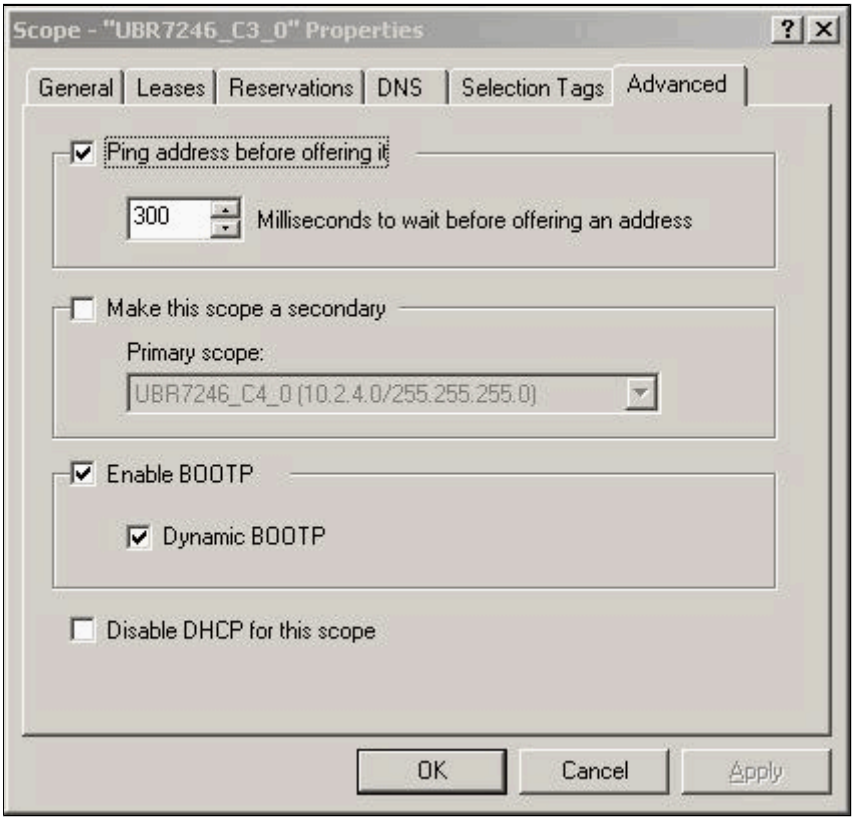
```
09/01/2000 12:19:33 name/dhcp/1 Info Protocol 0 04935 R3:
                    ->  siaddr = 0.0.0.0
09/01/2000 12:19:33 name/dhcp/1 Info Protocol 0 04935 R3:
                    ->  giaddr = 0.0.0.0
09/01/2000 12:19:33 name/dhcp/1 Info Protocol 0 04935 R3:
                    ->  chaddr = 0:10:a4:ff:61:8e
09/01/2000 12:19:33 name/dhcp/1 Info Protocol 0 04935 R3:
                    ->     dhcp-message-type =  3 request
09/01/2000 12:19:33 name/dhcp/1 Info Protocol 0 04935 R3:
                    ->     dhcp-client-identifier = 1 0 16 164 255 97 142
09/01/2000 12:19:33 name/dhcp/1 Info Protocol 0 04935 R3:
                    ->     ethernet? = 0:10:a4:ff:61:8e
09/01/2000 12:19:33 name/dhcp/1 Info Protocol 0 04935 R3:
                    ->     dhcp-requested-address = 10.200.68.201
09/01/2000 12:19:33 name/dhcp/1 Info Protocol 0 04935 R3:
                    ->     dhcp-server-identifier = 10.200.68.17
09/01/2000 12:19:33 name/dhcp/1 Info Protocol 0 04935 R3:
                    ->     hostname = "dell-port-PC"
09/01/2000 12:19:33 name/dhcp/1 Info Protocol 0 04935 R3:
                    ->     dhcp-parameter-request-list = 20
09/01/2000 12:19:33 name/dhcp/1 Info Protocol 0 04935 R3:
                    ->         1    subnet-mask
09/01/2000 12:19:33 name/dhcp/1 Info Protocol 0 04935 R3:
                    ->         3    routers
09/01/2000 12:19:33 name/dhcp/1 Info Protocol 0 04935 R3:
                    ->         15   domain-name
09/01/2000 12:19:33 name/dhcp/1 Info Protocol 0 04935 R3:
                    ->         6    domain-name-servers
09/01/2000 12:19:33 name/dhcp/1 Info Protocol 0 04935 R3:
                    ->         44   netbios-name-servers
09/01/2000 12:19:33 name/dhcp/1 Info Protocol 0 04935 R3:
                    ->         46   netbios-node-type
09/01/2000 12:19:33 name/dhcp/1 Info Protocol 0 04935 R3:
                    ->         47   netbios-scope
09/01/2000 12:19:33 name/dhcp/1 Info Protocol 0 04935 R3:
                    ->    vendor-encapsulated-options = 55 2 0 0
09/01/2000 12:19:33 name/dhcp/1 Info Protocol 0 04935 R3:
                    ->    end
09/01/2000 12:19:33 name/dhcp/1 Info Protocol 0 04935 R3:
                    ->    sname = ""
09/01/2000 12:19:33 name/dhcp/1 Info Protocol 0 04935 R3:
                    ->    file = ""
09/01/2000 12:19:33 name/dhcp/1 Info Protocol 0 04935 R3:
                    ----- END OF RECEIVED -- R3 -----
09/01/2000 12:19:33 name/dhcp/1 Info Protocol 0 05005 10.200.68.202
                    Offer to Host: dell-port-PC CID: 1:00:10:a4:ff:61:8e packet
                    'R3' was rejected in favor of an offer from another server.
```

Is There Another Host On the Same Network Already Configured With the IP Address That Is Going to be Offered to the Client?

Before it offers a lease, CNR can perform a ping to the IP address that it is about to offer. If it receives a positive answer, then it deactivates that lease and chooses a new IP address to offer, to avoid duplicate addresses on the network. By default, this behavior is disabled, but you can enable it on a per-scope basis from the GUI.

Choose **Scope > Properties** and click the **Advanced** tab; then check the **Ping address before offering it** checkbox:

Figure 5 – Scope Dialog Box



Conversely, you can issue this command from the CLI:

```
nrcmd>  scope name enable ping-clients
```

Example

If, on scope UBR7246_C4_0, you want to ping an address before it is offered, then issue this command:

```
nrcmd> scope UBR7246_C4_0 enable ping-clients

100 OK
ping-clients=enabled
```

This sample output shows the debugs in this situation:

```
09/01/2000 12:52:26 name/dhcp/1 Warning Protocol 0 0467
                    Unexpected ping reply received for AVAILABLE lease
                    '10.200.68.201' - it is being marked  UNAVAILABLE
```

NetPro Discussion Forums - Featured Conversations

Networking Professionals Connection is a forum for networking professionals to share questions, suggestions, and information about networking solutions, products, and technologies. The featured links are some of the most recent conversations available in this technology.

NetPro Discussion Forums - Featured Conversations for Network Management
Network Infrastructure: Network Management
Device Fault Manager Error (Patch IDU 1.2.5) - Sep 15, 2004 CPU reports - Sep 15, 2004 Netflow Report Over Time - Sep 15, 2004 Improve the Cisco Management Console - Sep 15, 2004 Password Problem Installing IPM 2.5 Database Server not running - Sep 15, 2004

Virtual Private Networks: Network and Policy Management
Cisco Works SNMP write community string Incorrect - Sep 14, 2004 nGenius Packet Analyzer - Sep 13, 2004 VPN/Security Management Solution License Issue - Sep 13, 2004 syslog message - Sep 11, 2004 Authentication Active Directory is failing - Sep 10, 2004

Related Information

- **Cisco CNS Network Registrar 6.1**
- **Configuring Network Registrar**
- **Cisco Network Registrar for the Cisco uBR7200 Series Universal Broadband Routers**
- **Cisco CNS Network Registrar Documentation**
- **CNR Frequently Asked Questions**
- **How to Recover a Corrupted Cisco Network Registrar Database**
- **Changing Settings for CNR Server Logs**
- **Technical Support - Cisco Systems**