INTERNET

Internet Naming and Address Management

By Eddie Ho and Greg Althaus

AIX Version 4 has built-in support for Dynamic Host Configuration Protocol (DHCP) and Dynamic Domain Name Service (DDNS) enhancements. These enhancements help to automate IP addresses and other tasks of the Domain Name Server, while eliminating downtime and costly diagnostic work.

Dynamic Host Configuration Protocol (DHCP) with the Dynamic Domain Name Service (DDNS) extension provides a way to automatically track and verify IP addresses. The increasing demand for connections to the Internet makes this essential.

DHCP

DHCP automatically assigns IP addresses and configuration parameters from a data pool of addresses that reside on a DHCP server. DHCP can also provide configuration parameters, such as default gateway, name servers, X-Windows parameters, and other user-defined values.

Traditionally, as users move and connect a new system to a TCP/IP network, the user is given a new IP address, default gateway, name server, and other required parameters. Users then manually enter these parameters to the TCP/IP configuration file and restart the protocol stack in the new location. With DHCP in AIX, human intervention is not necessary. It is as simple as "pack-and-move."

Three Internet Engineering Task Force (IETF) Request-For-Comments (RFC) documents define the architecture:

- **RFC 1533:** DHCP provides a framework for passing configuration information to hosts on a TCP/IP network. This memo describes the DHCP options and Bootstrap Protocol (`BOOTP`) vendor extensions.
- **RFC 1541:** Framework for passing configuration information to hosts on a TCP/IP network. DHCP is based on `BOOTP`, adding the capability of automatic allocation of reusable network addresses and additional configuration options.
- **RFC 1542:** Redefinition of `BOOTP` relay agents in the DHCP environment. This memo attempts to clarify and strengthen the specification in these areas.

The solution framework is a two-tiered client/server computing model. The client must be DHCP enabled, which can be either AIX Version 4 or PCs with TCP/IP support and DHCP client functionality. The client enters a network in an initializing state and broadcasts a discover message on the IP network. The message is then relayed by the `BOOTP` relay agent, and eventually delivered to the DHCP server for processing. The DHCP server responds to the client request with an offer message that consists of an available IP address and configuration information. The client then binds the IP address to the TCP/IP stack and restarts the protocol initialization.

The DHCP client acknowledges the DHCP server, which in turn updates the DNS server accordingly. AIX Version 4 supports both the client and the server implementations with DNS synchronization. Figure 1 summarizes multiple IP networks with DHCP and DDNS serving capabilities.

Protocol Flow between DHCP Client and Server

DHCP is an application-layer protocol that allows services to be solicited and negotiated from one or more servers. The protocol framework is BOOTP, which runs on User Datagram Protocol (UDP).

If network size requires cross-network DHCP serving, the inter-network router must support the BOOTP relay agent. This relay agent, responsible for the BOOTP message passing between two IP networks, is required if the DHCP clients and server are not in the same network. Cross-network DHCP serving can potentially cause massive broadcast storms if the network size is too large; in this case, careful design with one or more DHCP servers is recommended. The AIX implementation for both client and server portions consists of the following modules:

- Client daemon (dhcpcd)
- Server daemon (dhcpsd)
- Relay agent daemon (dhcprd)

Figure 2 shows a simple flow sequence for an assignment.

Shopping for an IP address begins from the DHCP client. The client broadcasts a message (DHCPDISCOVER) to all DHCP servers soliciting for service. This packet contains the client's connection requirements based on the current environment. All DHCP servers can receive DHCPDISCOVER broadcasts. One or more DHCP servers can respond with unicast or broadcast with IP address offer (DHCPOFFER).

The client receives all offers and determines those most desirable using the built-in algorithm. Usually the best match for suggested options will win. The client then broadcasts its response to the DHCPOFFER using a DHCPREQUEST packet. The target DHCP server will send a DHCPACK to acknowledge service rendered. Subsequently, all other servers that are not chosen will free up their address and return to listening mode for the next DHCPDISCOVERS packet.

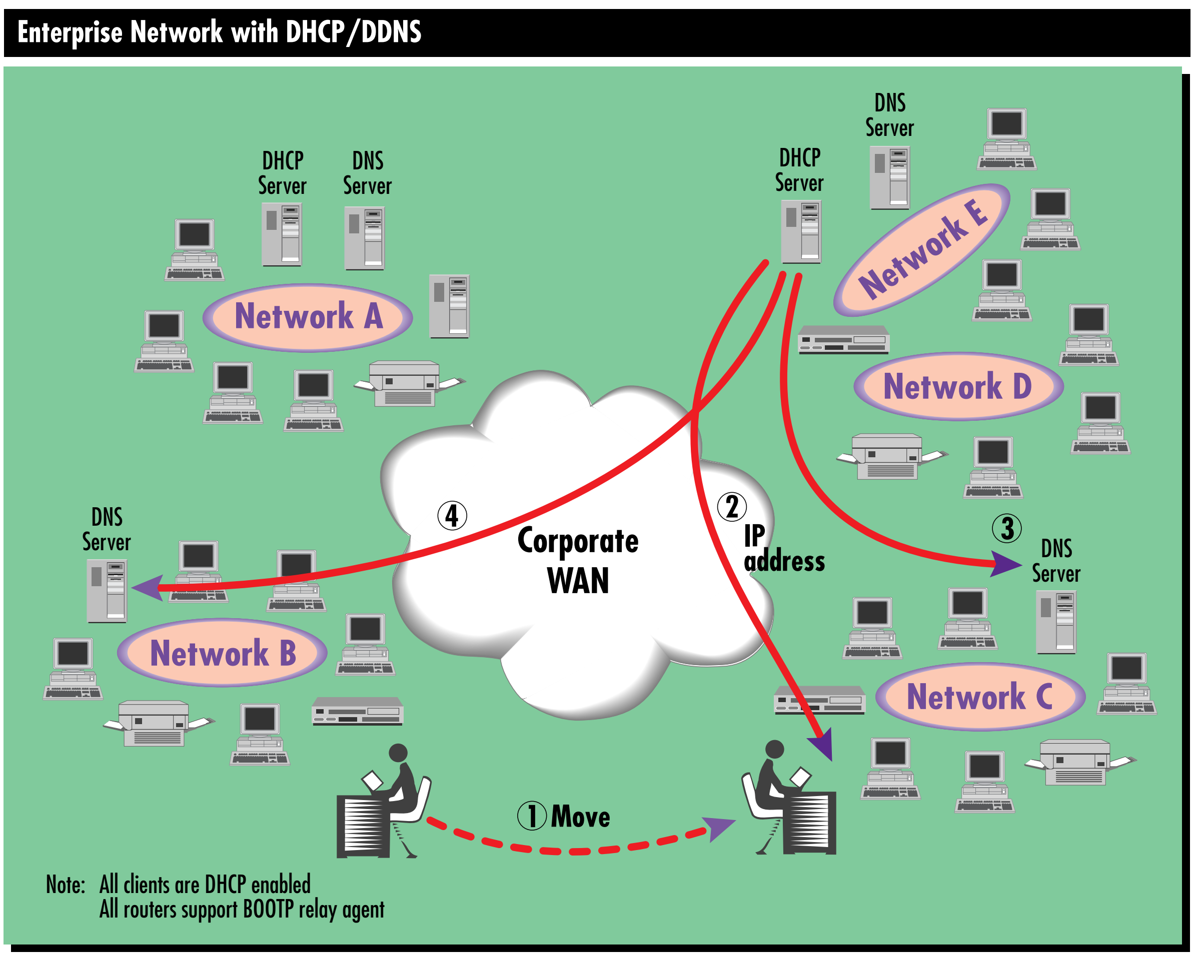

Figure 1. IP networks with DHCP and DDNS

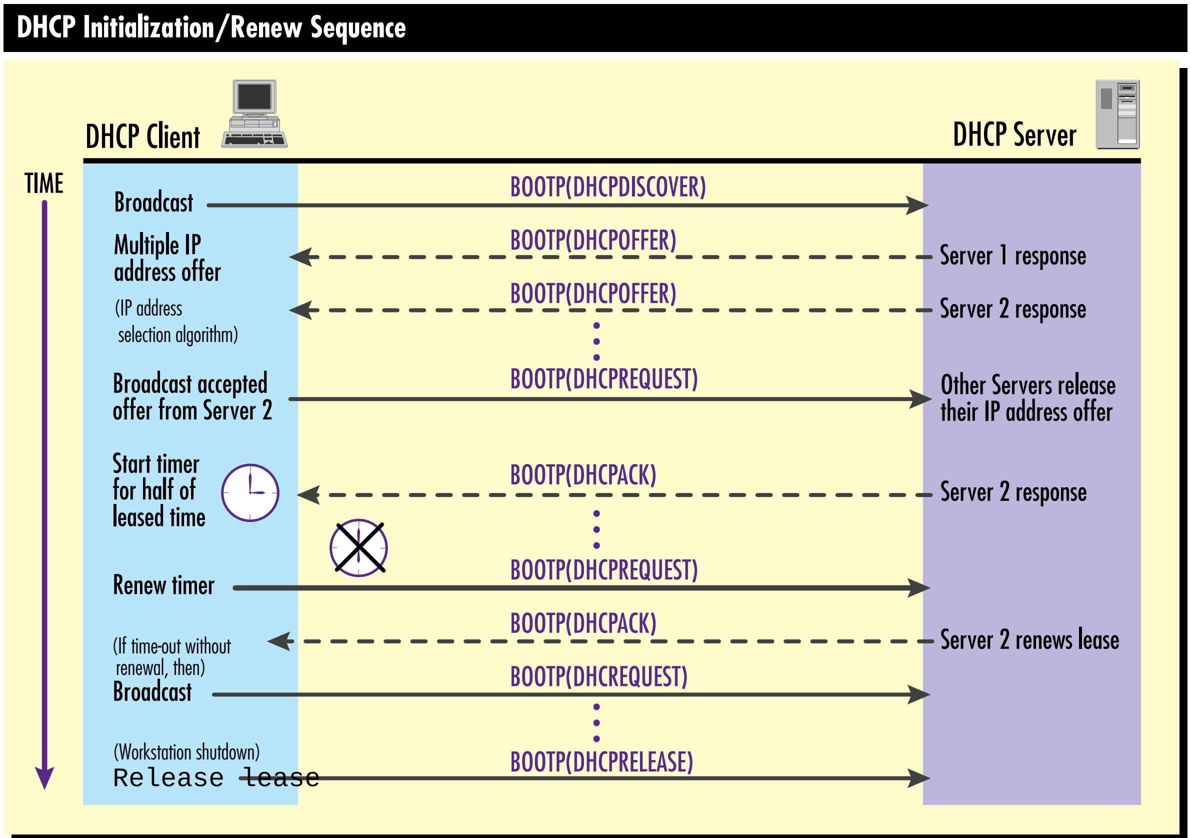

Figure 2. Flow sequence for an assignment

The client IP address is leased from the server and must be renewed periodically; therefore, a timer is started by the client and set up for half of the lease time. When it expires, the client sends a `DHCPREQUEST` packet directly to the server for renewal. If the client receives no response, the client will wait up to three fourths of the leased time, then broadcast a `DHCPREQUEST` packet soliciting for the new server. This ensures that the DHCP service continues without interruption.

AIX 4.1 DHCP/DDNS

The RISC System/6000 (RS/6000) DHCP design for both client and server conforms to all existing available RFCs. It is also tested for interoperability with other DHCP clients or servers in the marketplace. It currently supports LAN-based protocol, including Ethernet™, Token-Ring, and Fiber-optic Data Distribution Interface (FDDI) interfaces.

Figure 3 shows the latest interoperability list of confirmed products supported by the RS/6000. In fact, any product that supports the above RFCs should be able to interoperate without major problems. The DDNS implementation is based on the latest draft of RFC that is pending for approval. Few products in the marketplace offer both DHCP and DDNS capabilities; but none are based on the proposed draft. The AIX version will conform to the final draft when available.

Interoperability Products Supported by RS/6000

DHCP Server	DHCP Clients
AIX 4.2	MacOS, Windows 95, Windows NT 3.5, Windows 3.1, Chameleon, FTP Software
AIX 4.1.4	OS/2 Warp™ Connect
OS/2 Warp Server Sun FTP Software JOIN by Competitive Automation	AIX 4.2/AIX 4.1.4

Figure 3. Interoperability products supported by RS/6000

Configuration of the DHCP support can be grouped into three independent areas: server, client, and the BOOTP relay agent. Each RS/6000 can be configured for one of the three portions.

Configuring the DHCP Server

The DHCP server generates and offers IP addresses based on a set of predefined attributes or a current network service environment. The definition service is done by editing a flat ASCII file or using the Graphical User Interface (GUI). Based on BOOTP protocol and used in an Xstation environment, the DHCP server can assign Xstation terminal IP addresses and host name; it will be considered a permanent lease. The configuration process defines three major areas:

- Global resources for all networks
- Various resources for each network
- Resources for specific clients or sets of clients

Keys represent client information within the DHCP server. The first key is the client's *position* in the network, which links to the selected address pool for a candidate address. A client can be a member of a set machine that has similar features designated as a *class*. This class specification allows the server to provide extra options or features for a client.

Another key is the client ID, which can be a TCP/IP hostname or the Media Access Control (MAC) address. The server can use the client ID to provide personalized options. Based on these keys, the DHCP server can generate a reply per client request to serve the community of users. This enables the server to assign an address based on the network segment to which the user belongs.

If the class information is specified by the client, the default services for that class can be returned rapidly. For example, the accounting class can represent users in an accounting department, which needs access to a specific high-quality printer. Global resources provide overall consistency and baseline functions for each client. They are sent if no network, class, or client options override them. The hierarchy of options are client, class, network, and global options.

Figure 4 shows an example of the Ace Corporation, which has six IP networks. Network Information Center (NIC) has assigned a class C and a class B network address to Ace. The corporate users can be grouped into either a dynamic or static environment.

The class C network is used by a dynamic group of users in the marketing and sales organization. Each office within the building has an interface port and serves as a temporary mobile docking station for many sales personnel. In this example, a pool of dynamic users shares each office.

The class B network is used and shared by the remaining corporation in five static functional workgroups, including accounting, research and development, manufacturing, product testing, and the server network. Each of these five groups has a subnet of the class B network. Figure 4 shows the assignment of the network address and network mask.

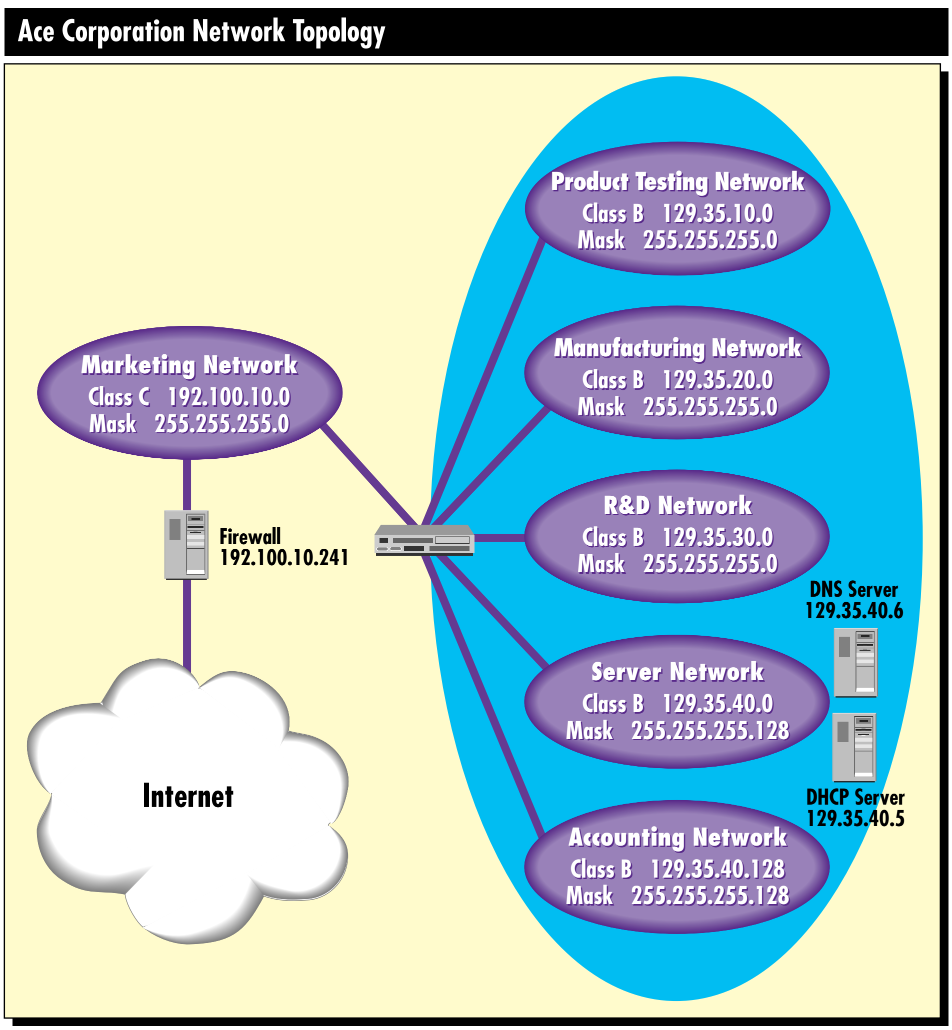

Figure 4. Network address and network mask assignment

```
network 192.100.10.0 192.100.10.2-192.100.10.240
{
   option 1    255.255.255.0            # Subnet mask for class C
   option 3    193.100.10.1             # Default gateway/router address
   option 6    129.35.40.5              # Default DNS Server
   option 51   2 hours                  # lease time of 2 hours since the sales
                                        # people are only around for an hour
   option 15   “marketing.ace.com”      # Default domain
}
```

Figure 5. Configuration for Class C marketing network

```
network 129.35.0.0 24                     # NIC assigned class B addr 129.25.0.0
                                          # 24 bit subnet mask is used
{
   option 1    255.255.255.0              # subnet mask for class B
   subnet      129.35.10.0                # subnetwork address
   {                                      # all addresses are assignable
      client 0 0 129.35.10.1              # address 129.35.10.1 is removed for
                                          # the router to use in Ace Corp.
      option 3   129.35.10.1              # default gateway/router address
      option 6   129.35.40.5              # default DNS server address
      option 15  “producttest.ace.com”    # default domain
   }
   subnet      129.35.20.0   129.35.20.2-129.35.20.200
   {                                      # subnetwork address with defined range
      option 3   129.35.20.1              # default gateway/router address
      option 6   129.35.40.5              # default DNS server address
      option 15  “manufacturing.ace.com”  # default domain
   }
   subnet      129.35.30.0   129.35.30.2-129.35.30.215
   {                                      # subnetwork address with defined range
      option 3   129.35.30.1              # default gateway/router address
      option 6   129.35.40.5              # default DNS server address
      option 15  “R&D.ace.com”            # default domain
   }
}
```

Figure 6. Configuration definitions

The class B network is partitioned into four class C networks. Both the Server network and the accounting group can share half of class C because of the organization size.

Figure 5 shows the configuration definition for the class C marketing network.

The network statement has two parameters. The first parameter is network address; the second indicates the range of valid addresses that can be assigned. If the second parameter is not specified, this implies that all addresses are assignable. The range is from 2 to 240 because 1 is used by the router in Ace and should not be assigned; the address above 240 is reserved for future static users due to the business environment.

Figure 6 shows configuration definitions for three of the class B networks—Product Testing, Manufacturing, and Research and Development.

Figure 7 shows the definitions for the Server and Accounting networks.

Figure 8 shows those global parameters responsible for the overall operations.

The client statement provides options for a specific client, such as reserving an IP address or removing an address from the range. When using `<string>` in this statement, all client `<string>` must be unique. Figure 9 shows the syntax.

The result of these parameters can have different effects and can be best summarized as shown in Figure 10.

```
network 129.35.0.0 25                   # NIC assigned class B addr 129.35.0.0
                                        # 25 bit subnet mask is used because
                                        # the 256 addresses are shared by 2
                                        # subnetworks.
{
    option 1      255.255.255.128       # subnet mask for class B
    subnet        129.35.40.0    129.35.40.64-129.35.40.126
                                        # subnetwork address with defined range
    {
       option 3    129.35.40.1          # default gateway/router address
       option 6    129.35.40.5          # default DNS server address
       option 15   "netserver.ace.com"  # default domain
    }
    subnet        129.35.40.128         # subnetwork address
    {
       option 3    129.35.40.129        # default gateway/router address
       option 6    129.35.40.5          # default DNS server address
       option 15   "accounting.ace.com" # default domain
       client 0 0  129.35.40.129        # client addr 129.35.40.129 is removed
       client 1 0x1005ACABADAE 129.35.40.130 # IP address 129.35.40.130 is reserved
                                             # for Ethernet @ 0x1005ACABADAE
    }
}
```

Figure 7. Definitions for Server and Accounting networks

```
supportunlistedClients       yes     # support all clients without
                                     #  explicitly listing in this file
supportBOOTP                 yes     # Ace has Xstation and network
printer
                                     #  that uses BOOTP protocol
leasetimedefault             5 days  # 5 days lease time for IP address
leaseexpireInterval          1 day   # time for the DHCP server to recover
                                     # lost IP address due to end user not exit
                                     #  normally when disconnect
```

Figure 8. Global parameters

```
client   <hardware type>   <hardware address>   <IP address>
         1=Ethernet        address <string>     <none>
         6=Token Ring                           <any>
         1=FDDI
         0=<string> on the next field
```

Figure 9. Syntax of client statement

Summary of Parameters

Hardware Type	Hardware Address	IP Address	Results
0	0	<IP address>	`<IP address>` will be removed from the pool
0	<string>	<IP address>	Client whose identifier matches this `<string>` will be assigned `<IP address>`; for example, `<string>` can be people's title and is used for assignment
<type>	<address>	<IP address>	Specific hardware type with the hardware address will be assigned the special IP address
<type>	<address>	none	Do not respond for this type with specified address
0	<string>	none	Do not respond whenever this string is specified
<type>	<address>	any	Give any IP address to this type with specified address. Used when `supportunlistedclients=no`
0	<string>	any	Give any IP address if this string is specified. Used when `supportunlistedclients=no`

Figure 10. Parameter results

```
smit tcpip -> Further Configuration -> Server Network Services ->
Other Available Services -> dhcpsd Subsystem
```

Figure 11. SMIT sequence for starting the DHCP server

Server configuration consists of setup and control; setup consists of initialization of all DHCP parameters for all potential clients. The configuration data is stored in the `/etc/dhcpsd.cnf` file. The auto start of the server is in `/etc/rc.tcpip`, or it can be started from the following `<smit>` sequence. Figure 11 shows the selection order.

The command to start the GUI server definition is `<dhcpsconf>`. See Figure 12.

The GUI interface can mask the syntax complexity of the configuration file and simplify the setup. The menu bar contains pull-downs that define the server default features, such as leased time duration, IP address cleanup, and so on. The panel has three main working areas:

- **Option List:** Descriptive selectable options that the server can support
- **Key List:** Entries to describe a client, including network position, class, and client ID
- **Main Window List:** Logical view of the options for each key

Configuring the DHCP Client

The goal of the DHCP client is full mobility without reconfiguration for each connection. This is done by defining a set of client preferences—the existing network and server environment requirements. The intent for the DHCP server is to provide a similar running environment in the destination network. The client setup is done using System Management Interface Tool (SMIT) panels.

The client broadcasts the preference information during the connection phase. One or more DHCP servers will examine the preference (preference list is optional for a client) and provide a counteroffer. The client will choose the intended server based on the best-matched algorithm. The client can specify only the types of information that belong to two major categories:

- TCP/IP network related, including static routes, NetBIOS name server, interface Maximum Transmission Unit (MTU) size, and so on
- Server services related, including Time Server, Print Server, DNS, NIS servers, and so on

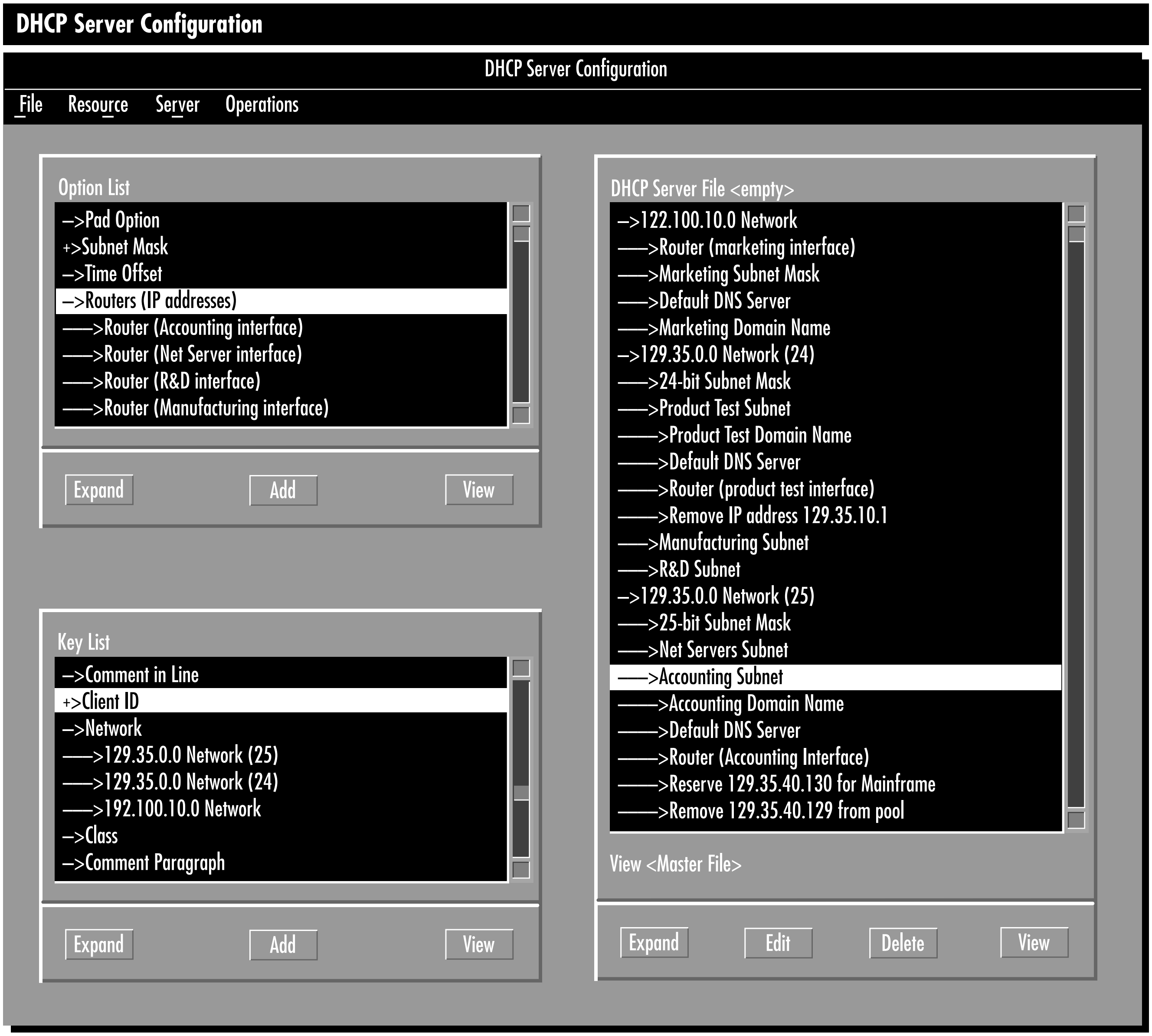

Figure 12. DHCP server configuration

The setup information is stored in two files:

`dhcpcd.ini` DHCP configuration file consists of directives that can be specified for a client; the preferred option list

`/etc/rc.net` Information for the startup interface

The SMIT configuration sequence for a client is as follows:

```
smit tcpip   Use DHCP for TCP/IP
             Configuration & Startup
```

SMIT can also be used for client daemon control using the sequence as follows:

```
smit tcpip → Further Configuration →
Server Network Services →
        Other Available Services →
dhcpcd Subsystem
```

BOOTP Relay Agent

A router usually does the relay when routing BOOTP broadcast from one subnet to another. AIX can also do TCP/IP routing with support for BOOTP forwarding. Using a method similar to the one used in the DHCP server, the BOOTP relay agent is started and stopped.

The configuration is done by adding a line to the `/etc/dhcprd.cnf` file. Multiple servers can be added to the configuration file. The relay agent will send the incoming packet to all servers that are defined in this file.

Storage Requirements

DHCP and DDNS server can be set up in a single server because of the traffic flow pattern and synergy in functions. Storage requirements for the server are dependent on the number of clients. Disk usage for each client is 360 bytes; therefore, a DHCP server with 10,000 addresses requires 3,600,000 bytes of storage space. Memory utilization is the same as storage since the entire database will be loaded due to performance considerations.

Conclusions

DHCP and DDNS functions can automate the labor-intensive process of address and resource control in a large network environment. This architecture is still emerging and extensions are needed to integrate with the Point-to-Point Protocol (PPP) for remote and mobile users. Also, server availability must be addressed in a 24x7 environment. Based on the current design point, whenever a DHCP server is unavailable, new DHCP clients cannot join the network and existing DHCP clients cannot renew their leases. Another area that needs focus is the DHCP database technology in a distributed environment, which can directly affect the server performance, redundancy, and management issues.

Eddie Ho, IBM Corporation, 11400 Burnet Road, Austin, TX 78758. Mr. Ho is a senior consulting marketing representative in the RS/6000 Executive Briefing Center. He has a BS in Computer Science from the University of Wisconsin and an MS in Computer Science from North Dakota State University.

Greg Althaus, IBM Corporation, 11400 Burnet Road, Austin, TX 78758. Mr. Althaus is a staff programmer in the AIX Open Systems Communication Development. He has a BS in Computer Science from the University of Texas in Austin.