4

DYNAMIC HOST CONFIGURATION PROTOCOL

4.1 INTRODUCTION*

In the early days of the Internet's existence, when hosts numbered in the hundreds, assigning an IP address to a device was fairly trivial. It was simply one of the configuration parameters entered manually on each host. This "once and done" or *static* address assignment process using a hard-coded IP address certainly was simple, but it inhibited the host's mobility among different networks or subnets. Enabling mobility required the cumbersome task of reconfiguring the host with a new IP address based on the present location or network to which connection was desired.

Nonetheless, you will likely have a set of static addresses for devices on your network that do not require mobility, such as routers, servers, IP PBXs, and so on. It's imperative to keep track of which IP addresses on allocated subnets are statically assigned, which are assigned to address pools for dynamic assignment, and which are free or reserved for future use. Maintaining a subnet IP inventory is a recommended practice to maintain a record of addresses assigned on each subnet to minimize duplicate or otherwise erroneous IP address assignments. Just make sure the inventory of static

* Initial sections of this chapter are based on Chapter 3 of Ref. 11.

IP Address Management: Principles and Practice, by Timothy Rooney

addresses matches what's actually been provisioned on the router, server, or statically addressed device. Performing periodic baselines of address assignments through various forms of discovery or ping sweeps can help identify any mismatch as we'll discuss in Chapter 14.

4.2 DHCP OVERVIEW

The Dynamic Host Configuration Protocol (DHCP) is a client–server protocol for devices connecting to an IP network to automatically obtain an IP address. DHCP has been a tremendous time saver for IP network administrators. It enables a device to broadcast its request for an IP address, and have one or more DHCP servers within the IP network service the request without user intervention. For most end user devices such as laptops, VoIP phones, PDAs, and others, the DHCP process transpires "behind the scenes" upon device boot-up or connection to a wire line or wireless network without user intervention. DHCP also enables efficient use of IP addresses by allowing an IP address to be reused among devices within dynamically allocated address pools. A given IP address may be used by one device one day and a different device the next.

DHCP is supported as part of both IPv4 and IPv6. We'll discuss the IPv4 version in this chapter and the IPv6 version in the next. When we use the term "DHCP" in this chapter, we're referring to the IPv4 version. Defined in RFCs 2131 (32) and 2132 (33) with many additions in subsequent RFCs*, DHCP is built on the foundation of an older protocol, the Bootstrap Protocol, referred to as BOOTP. BOOTP, initially specified in RFC 951 (34), provides automation of address assignment but is restricted to preassigning a given IP address to a particular device, identified by its network interface (MAC) address. Thus, a BOOTP server is configured with a list of MAC addresses and corresponding IP addresses. DHCP incorporates this functionality with the added capability of assigning IP addresses to clients without requiring *a priori* knowledge of each client's hardware address. In effect, DHCP supersedes BOOTP, enabling backward compatibility with BOOTP clients.

DHCP supports three types of IP address allocation:

1. *Automatic Allocation*. The DHCP server assigns a permanent IP address to the client.
2. *Manual Allocation*. Like BOOTP, the DHCP server assigns a "fixed" IP address based on the particular device's hardware address.
3. *Dynamic Allocation*. The DHCP server assigns an IP address for a limited time period, after which it can be reassigned, perhaps, to a different device.

Automatic allocation may be useful for a particular set of users or devices requiring a permanent IP address assignment via DHCP, where there's no requirement for a particular user or device to have a particular IP address. In other words, you may want to set aside a number of "permanent" addresses without directly associating each IP address

* Please refer to the RFC Index at the back of this book for a complete list.

with a particular hardware address. This is in contrast to Manual DHCP, which associates a particular hardware address with a corresponding IP address.

Dynamic allocation is commonly used to set up address pools in DHCP servers in order to "reuse" IP addresses. Under dynamic allocation, the DHCP server leases its IP addresses to clients for a fixed period of time. In this way, the DHCP server can assign an IP address to a particular client for a given time period referred to as the lease time, and when the IP address becomes available due to the expiration of the lease or the client relinquishing the address, reassign the same address to a different client. The lease time is a configurable parameter within the DHCP server implementation.

Regardless of the DHCP address allocation type, the process by which a DHCP client obtains a lease is the same. The basic process begins with a DHCP client broadcasting a DHCPDISCOVER packet. Since the client has neither an IP address nor generally any information about the IP network, it inserts the all-zeroes address as the source address and the broadcast (all-ones) address as the destination address within the IP header. Let's assume that a DHCP server has been deployed on the same subnet to which the DHCP client is connected. Upon receiving the DHCPDISCOVER packet, the DHCP server will determine if it has an address available on this subnet on which the DHCPDISCOVER was received.

If an address is available in the pool, the DHCP server will send a DHCPOFFER packet to the client, offering an IP address and associated configuration parameters, called *options*. The DHCP client may receive more than one DHCPOFFER if multiple DHCP servers are servicing this subnet. The client will select one configuration set and broadcast a DHCPREQUEST packet, specifying the selected DHCP server whose offer it has accepted. The selected DHCP server will acknowledge the DHCPREQUEST with a DHCPACK once it has recorded the lease information in nonvolatile storage, thereby binding the IP address to the DHCP client. This basic message flow, illustrated in Figure 4.1 is sometimes referred to as the "DORA" process – Discover, Offer, Request, and Ack.

In this simple example, the DHCP server resides in the same subnet as the DHCP client. The client broadcasts the DHCPDISCOVER packet on the network. Since the DHCP server resides in the same network, it receives the broadcast and processes the packet. Knowing the network from which the broadcast originated, the DHCP server can assign an available IP address on the network. But do you have to deploy a DHCP server on every subnet? Fortunately, no; the DHCP server simply must be reachable from the subnet via the IP routing infrastructure. The router(s) receiving the DHCPDISCOVER broadcast packet will not propagate the broadcast, as this would create

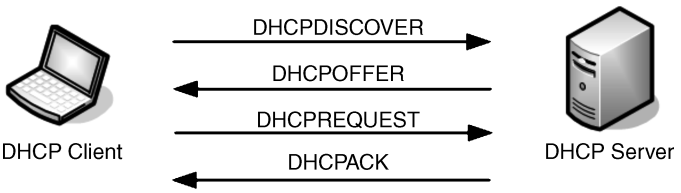


Figure 4.1. DHCP "DORA" process.

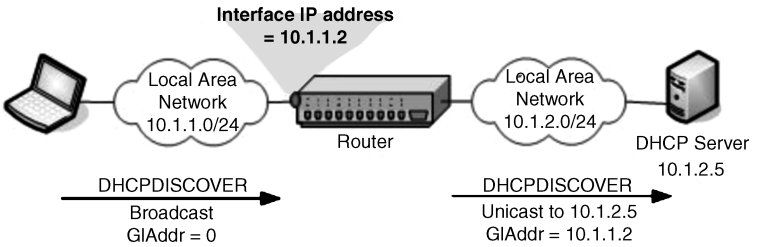


Figure 4.2. DHCP Relay (11).

excessive and needless IP traffic. Instead, the router will forward or *relay* the packet to the intended DHCP server(s). Each router configured to perform this relay function is referred to as a *relay agent*. Each relay agent must be configured with the IP addresses of each DHCP server serving the subnet. This configuration parameter, commonly referred to as the DHCP Relay address, enables the router to accept the DHCPDISCOVER broadcast, look up the DHCP server(s) configured for DHCP Relay, and then route the DHCPDISCOVER packet via unicast directly to each DHCP server as illustrated in Figure 4.2.

In the process, the router modifies the DHCPDISCOVER packet to insert the IP address of the interface on which the DHCPDISCOVER was received into the relay agent (gateway) interface address field. This parameter enables the DHCP server to identify the subnet on which an address assignment has been requested. Note that when the gateway interface address (GIAddr) field is zero, the DHCP server assumes the subnet on which to assign the IP address is the same as that on which the DHCPDISCOVER was received (via direct broadcast).

In addition to the four-packet exchange outlined above, the IETF has adopted RFC 4039 that defines a rapid commit option, option 80. This option is modeled after the DHCPv6 equivalent defined in the next chapter, and halves the messaging requirements by enabling the server to simply send a DHCPACK in response to a DHCPDISCOVER message. The client would include the rapid commit option in its DHCPDISCOVER message. Servers responding with an address assignment would directly issue an ACK packet, also including the rapid commit option. Rapid commit functionality is desirable particularly for mobility applications such as cell phones that have limited bandwidth available. Note that each server responding will assume the address it assigned is leased, so rapid commit should be used with either short lease times or for support by a limited number of servers if normally there are many serving the same subnet.

4.2.1 DHCP message types

We've introduced the four basic DHCP message types, so let's expand on this and review the complete set of DHCP messages and their respective meanings. We often omit the "DHCP" prefix on these messages and just capitalize the first letter, but here's how they're officially defined:

- *DHCPDISCOVER*. Issued from the client to the server to solicit DHCP address assignment; the DHCPDISCOVER may include parameters or options required by the client
- *DHCPOFFER*. Issued from the server to the client indicating an IP address offer including its corresponding lease time (and other configuration parameters) to the client in response to a DHCPDISCOVER.
- *DHCPREQUEST*. Issued from the client to a server in response to a DHCPOFFER to accept or reject the offered IP address, along with desired or additional parameter settings. The DHCPREQUEST is also used by clients desiring to extend or renew their existing IP address lease.
- *DHCPACK*. Issued from the server to the client to positively acknowledge the grant of the IP address lease and associated parameter settings. The client may now begin using the IP address and parameter values.
- *DHCPNAK*. Issued from the server to the client to negatively acknowledge the DHCP transaction. The client must cease the use of the IP address and reinitiate the process if necessary.
- *DHCPDECLINE*. Issued from the client to the server, to indicate that the IP address offered by the server is already in use by another client. The DHCP server will then typically mark the IP address as unavailable.
- *DHCPRELEASE*. Issued from the client to the server to inform the server that the client is relinquishing the IP address. The client must cease the use of the IP address thereafter.
- *DHCPINFORM*. Issued from the client to the server to request non-IP address configuration parameters from the server. The server will formulate a DHCPACK reply with the associated values as appropriate.
- *DHCPFORCERENEW*. Issued from the server to the client to force a client into the INIT state* in order to obtain a new (different) IP address. Few clients have implemented support of this message.
- *DHCPLEASEQUERY*. Issued from a relay agent or other device to a server to determine if a given MAC address, IP address, or client-identifier value has an active lease and its associated lease parameter values according to the DHCP server (used primarily by broadband access concentrators or edge devices).
- *DHCPLEASEUNASSIGNED*. Issued from a server to a relay agent in response to a DHCPLEASEQUERY informing the relay agent that this server supports that address but there is no active lease.
- *DHCPLEASEUNKNOWN*. Issued from a server to a relay agent in response to a DHCPLEASEQUERY informing the relay agent that the server has no knowledge of the client specified in the query.
- *DHCPLEASEACTIVE*. Issued from a server to a relay agent in response to a DHCPLEASEQUERY with the endpoint location and remaining lease time.

* We'll discuss DHCP states next.

RFC 2131 defines a number of states in which the client may exist with respect to its IP address configuration using DHCP. The following states are defined:

- *INIT*. Initialization, meaning the client has neither an IP address nor any prior configuration information.
- *INIT-REBOOT*. The client initializes, though it has prior IP address information, and desires to confirm its settings.
- *BOUND*. The client and server are bound to their IP lease agreement.
- *RENEWING*. The client is attempting to renew the lease.
- *REBINDING*. The client is approaching lease expiration and is attempting to renew the lease.
- *SELECTING*. Intermediate state where the client is awaiting and evaluating DHCPOFFERs from DHCP server(s).
- *REQUESTING*. Intermediate state where the client has selected an Offer and wishes to accept it or has identified an Offer for an IP address that is already in use, in which case it sends a DHCPDECLINE to the server.
- *REBOOTING*. Client is attempting to rebind after a reboot.

The distinction between renewing and rebinding boils down to the urgency of the renewal request, with rebinding being of higher urgency, and to transport mode, with renewals being unicast and rebinding being broadcast. When a lease is initially obtained, the client sets two timers:

- T1 = 50% of the lease time by default.
- T2 = 87.5% of the lease time by default.

These timer values may be modified upon agreement between the DHCP client and the server by specifying values within corresponding options within the DHCP packet exchange. Upon expiration of the T1 timer, the client enters the renewing state and attempts to renew the lease by unicasting a DHCPRENEW message to the DHCP server from which it obtained the lease. If a DHCPACK is received, the client re-enters the bound state. If a DHCPNAK is received, the client ceases the use of the IP address and enters the INIT state. Otherwise, having not received a response, the client awaits the expiration of the T2 timer, then broadcasts a DHCPRENEW in an attempt to renew the lease. The broadcast is issued in case the original server from which the lease was obtained is down and a failover DHCP server is available to renew the lease.

Figure 4.3 depicts the state transition diagram among these DHCP client states and the respective state transition mechanisms. Note that the DHCPINFORM message is not included in the figure as this relates to non-IP address parameters. A client already configured with an IP address issues the DHCPINFORM to request additional parameter settings and the server replies with a DHCPACK indicating the requested values.

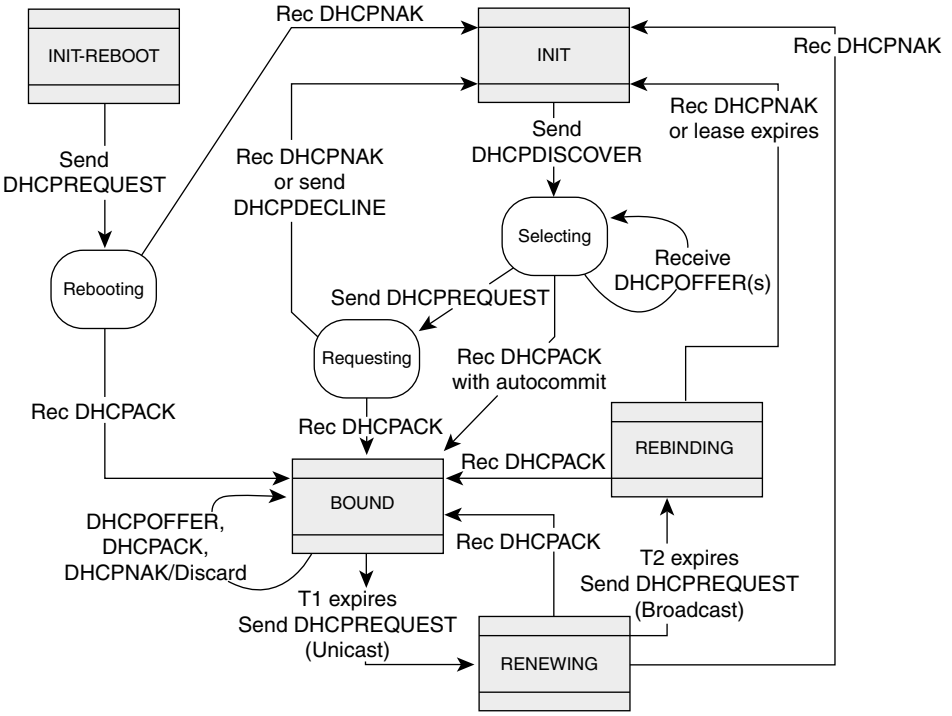


Figure 4.3. DHCP state transitions (32).

4.2.2 DHCP Packet Format

Let's examine the fields in the DHCP packet* and how they relate to the overall DHCP process. Figure 4.4 displays the field layout, and we'll describe each field following the figure.

DHCP packet field descriptions:

- *Operation Code*. Leveraging the BootP predecessor, the values for this field are
  - 1 = BootRequest
  - 2 = BootReply

Note that the type of DHCP message (Discover, Offer, Request, etc.) is actually defined in the options field with option number 53, DHCP message type, with the following valid values:

- 1 = DHCPDISCOVER
- 2 = DHCPOFFER
- 3 = DHCPREQUEST

* A "DHCP packet" is transported over IP within an IP packet. The DHCP "application" uses the packet header format described in this chapter.

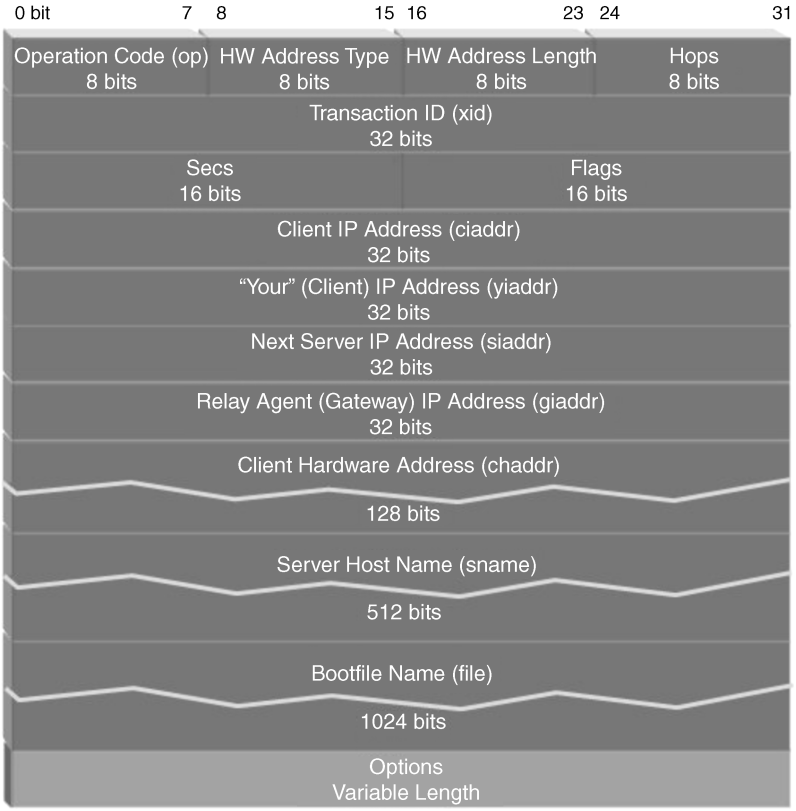


Figure 4.4. DHCP packet fields (32).

- 4 = DHCPDECLINE
- 5 = DHCPACK
- 6 = DHCPNAK
- 7 = DHCPRELEASE
- 8 = DHCPINFORM
- 9 = DHCPFORCERENEW
- 10 = DHCPLEASEQUERY
- 11 = DHCPLEASEUNASSIGNED
- 12 = DHCPLEASEUNKNOWN
- 13 = DHCPLEASEACTIVE

- *Hardware Address Type*. The type of hardware or MAC address, such as Ethernet, 802, and so on.
- *Hardware Address Length*. Defines the length of the MAC address in octets.

- *Hops*. Set to zero by clients, this field can be incremented by each router between the client and the server.
- *Transaction ID (xid)*. A random number chosen by the client to correlate messages and responses between the client and the server.
- *Seconds (secs)*. The number of seconds that have elapsed since the client began the process of obtaining an IP address or renewal.
- *Flags*. This field is used by DHCP clients that cannot receive unicast IP packets until its IP protocol software has been configured. For such cases, the client sets the first bit in this field to 1, and sets the remaining bits to 0. When set to 1, the server, if locally connected, or the relay agent will broadcast the Offer and Ack messages to the client; otherwise, the server or relay agent will send them to the unicast address specified in the yiaddr field. This bit is sometimes referred to as the broadcast bit within the Flags field.
- *Client IP Address (ciaddr)*. The IP address of the client used when known by the client, for example, when in the BOUND, RENEWING, or REBINDING state.
- *Your IP Address (yiaddr)*. The IP address assigned by the DHCP server for use by the client.
- *Server IP Address (siaddr)*. The IP address of the "next" server to use for bootstrapping as provided by the DHCP server.
- *Gateway Interface Address (giaddr)*. IP address of the interface on which the DHCP broadcast was received as populated by the relay agent.
- *Client Hardware Address (chaddr)*. The link layer or hardware address of the client provided by the client.
- *Server name (sname)*. DHCP server host name.
- *File*. Boot file name, null or fully qualified directory pathname.
- *Options*. Additional IP parameters such as lease time, domain name, default gateway, and subnet mask (see next section for a complete list). The first four octets of the options field are always the magic cookies of value (in hex): 63825363. This is a carryover from the original BootP specification in RFC 951 that was intended to provide a means to interpret the options, for example, for vendor-specific purposes.

4.3 DHCP SERVERS AND ADDRESS ASSIGNMENT

Each DHCP server can be configured with multiple address pools serving several different subnets in various locations. In fact, for some DHCP server implementations, the same address pool can be configured on multiple DHCP servers for redundancy. This will be discussed in more detail in Chapter 7. The DHCP server keeps track of the state of all IP addresses across all of its configured address pools. When an address is leased to a client, the server generally tracks not only the lease time for the IP address but also an identifier for the client leasing the IP address. This identifier is typically the layer 2

(MAC) address of the client, as obtained via the chaddr field, though the client identifier field, option 60, may also be used.

The use of the client identifier (client ID) option over the chaddr field was suggested to maintain an identifier for the device even if the link hardware is moved to another device. But in practice, most devices either do not provide a client ID or copy the value of the chaddr field into the client ID option.

The basic decision process typically used by DHCP servers in offering an address is based on the following:

- If the client has a leased address as recorded in the DHCP server, the server will assign this address.
- If the client previously had an address that is now expired or released but is still available, the server will assign this address.
- If the client includes an address in the Requested IP Address option, option 50, and the address is available, the server will assign this address.
- The server will assign an available address from a pool on the same subnet on which the DHCPDISCOVER broadcast was received if the GIAddr field is zero, or on the subnet indicated by the GIAddr value if nonzero. Additional criteria based on parameters within the DHCPDISCOVER packet may dictate from which pool the address gets assigned if there are multiple pools serving the subnet in question. These parameters are generically referred to as client class parameters and are discussed next.

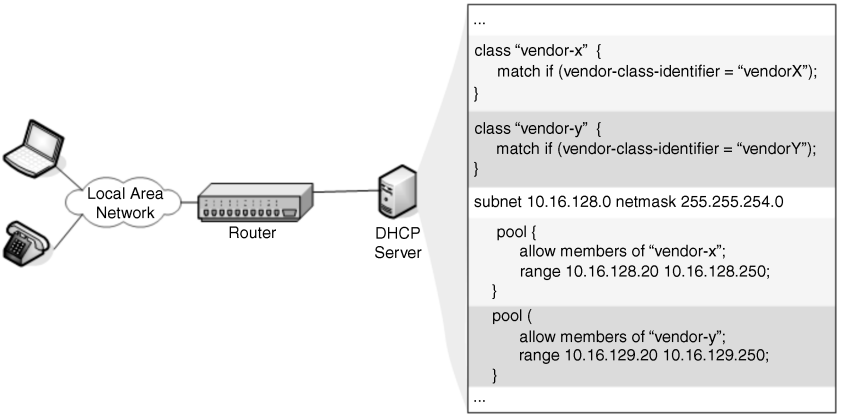
4.3.1 Device Identification by Class

Client class parameters provide a means both for the DHCP client to provide additional information to the DHCP server and for the DHCP server to recognize clients requiring unique IP address or parameter assignments. For example, you may want to dedicate one address pool for VoIP devices and a separate pool for data devices. This may be motivated by administrative concerns or by source routing policies for voice versus data packets from the respective devices. Most DHCP servers, including those available from the Internet Systems Consortium (ISC) and Microsoft, enable specification of vendor class or user class values to match on to provide such categorization. The DHCP server can be configured to associate a particular vendor class or user class value or set of values as criteria in assigning an address from an address pool.

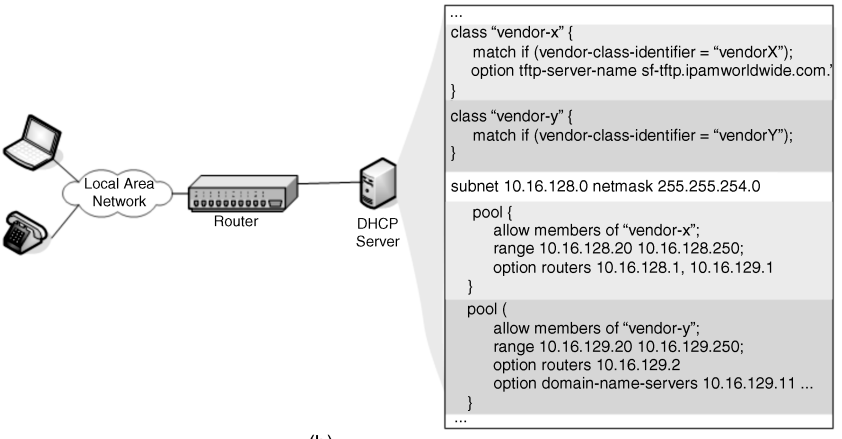
Let's consider an example. Recall in Chapter 3, in allocating address space for IPAM Worldwide, we allocated the subnet 10.16.128.0/23 for VoIP devices in San Francisco. Many organizations allocate a single subnet in a location and define two separate address pools for different VoIP device vendors due to differences in initialization and configuration requirements. In IPAM Worldwide's case, we'll need to define one address pool within the 10.16.128.0/23 subnet for VoIP devices for "vendor X" and a different pool within the same subnet for "vendor Y" VoIP devices. We then define two pools on IPAM Worldwide's DHCP server for each address pool, for example, the 10.16.128.20–10.16.128.250 address range and the 10.16.129.20–10.16.129.250

address range. We've shown them to be of equivalent size for simplicity, but there's no requirement that this be the case. In our IP subnet inventory, whether in a spreadsheet, database, or IP address management system, we can record these pools within these respective subnets. We should have also recorded static address assignments, for example, 10.16.128.1 for a router, 10.16.128.6 for a server, and so on.

In Figure 4.5a, we'd like to configure our DHCP server to discriminate between the VoIP phone vendors and assign addresses from different address pools. The first step is to determine what information in the DHCP packet can be used to uniquely identify each class of devices as VoIP phone or laptop. Typically, your VoIP phone provider will inform you that there is a particular string in the vendor class identifier option (option 60); let's say, originally enough this string is "vendorX" for vendor X and "vendorY" for vendor Y.



(a)



(b)

Figure 4.5. (a) Client classing example using DHCP configuration pseudocode based on Ref. [35]. (b) Specifying additional configuration information for DHCP clients by class based on Ref. [35].

We can define a class in the DHCP server for each vendor per the example in Figure 4.5a, though the syntax will depend on your DHCP server vendor (or IPAM tool). In this example, we configure the DHCP server to categorize devices sending DHCP packets with option 60 = "vendorX" as devices of the vendor-x class. Likewise, we define a class to discriminate vendor Y devices by defining the vendor class identifier option having a value of "vendorY" per the match-if clause in Figure 4.5a. In addition, a third pool could be set up as the "default" pool for clients not matching other defined client classes.

Now that we've defined our two classes, enabling the DHCP server to identify packets as originating from devices belonging to one class or the other, we can now instruct the server how to handle these requests. We can define two address pools within the respective subnet declaration, as we'd like to separate the address assignments for these two classes of devices. Within our 10.16.128.0/23 subnet, we define one address pool for vendor-x class devices as containing addresses 10.16.128.20–10.16.128.250 and a second address pool on the subnet for vendor-y class devices as containing 10.16.129.20–10.16.129.250, shaded to map to the class definition in the figure.

When configured per Figure 4.5a, the DHCP server will now examine each DHCPDISCOVER packet from devices on the 10.16.128.0/23 subnet to discern the class of device, then assign vendor X devices an address from the 10.16.128.20–10.16.128.250 pool, and assign vendor Y devices from the 10.16.129.20– 10.16.129.250 pool. Note that there may be additional parameters or option settings you may wish to define within each of these pool statements to provide configuration information according to each class of device, as we'll discuss a bit later.

Depending on the vendor of the DHCP server you deploy, there are various menu interfaces or text file editors that can be used for managing the configuration of address pools and server behavior as well as criteria you can specify to dictate address assignment logic. For example, Microsoft DHCP servers can be configured through a Windows graphical user interface (GUI), while ISC DHCP servers can be configured via text editor. However, the ISC DHCP provides more flexibility in defining client classes beyond user class and vendor class; any parameter in the packet can be examined and filtered upon for client class processing, including the chaddr field for MAC address filtering, or any other parameter present. For mixed ISC and Microsoft environments, the use of a centralized IPAM system could help abstract the individual vendor interfaces and enable configuration of both with a single interface.

4.4 DHCP OPTIONS

Clients can request settings for particular options, and servers can assign these and other parameters based on the DHCP server configuration. DHCP administrators can define groupings of options to be assigned to all or certain DHCP clients based on the client's hardware address, client class value, or other DHCP packet parameter.

As discussed in the previous section, we set up two client classes for IPAM Worldwide's San Francisco office corresponding to VoIP devices by vendor. Devices of these types will likely require different configuration parameters. For example, Cisco VoIP devices typically require option code 66 or 150, while Avaya VoIP devices require

TABLE 4.1. DHCP Options

Code	Name	Len	Meaning	Reference
0	Pad	0	None	RFC 2132 (33)
1	Subnet mask	4	Subnet mask in "IP address" format	RFC 2132 (33)
2	Time offset	4	Time offset in seconds from UTC (deprecated by RFC 4833 that specifies the use of options 100 and 101)	RFC 2132 (33)
3	Router	*N*	*N*/4* Router (default gateway) addresses	RFC 2132 (33)
4	Time server	*N*	*N*/4 Timeserver addresses	RFC 2132 (33)
5	Name server	*N*	*N*/4 IEN-116† name server addresses	RFC 2132 (33)
6	Domain server	*N*	*N*/4 DNS server addresses	RFC 2132 (33)
7	Log server	*N*	*N*/4 MIT Laboratory for Computer Science (LCS) UDP log server addresses	RFC 2132 (33)
8	Quotes server	*N*	*N*/4 "Quote of the day" server addresses	RFC 2132 (33)
9	LPR server	*N*	*N*/4 Line printer server addresses	RFC 2132 (33)
10	Impress server	*N*	*N*/4 Imagen Impress server addresses	RFC 2132 (33)
11	RLP server	*N*	*N*/4 Resource Location Protocol server addresses	RFC 2132 (33)
12	Hostname	*N*	Client hostname string	RFC 2132 (33)
13	Boot file size	2	Size of boot file in 512 byte blocks	RFC 2132 (33)
14	Merit dump file	*N*	File pathname to which the client should dump its core image in the event of a client crash	RFC 2132 (33)
15	Domain name	*N*	The DNS domain name of the client	RFC 2132 (33)
16	Swap server	*N*	Swap server address	RFC 2132 (33)
17	Root path	*N*	Path name for the client's root disk	RFC 2132 (33)
18	Extension file	*N*	Path name of a file containing vendor-extension information retrievable via TFTP	RFC 2132 (33)

(*Continued*)

TABLE 4.1. DHCP Options (*Continued*)

Code	Name	Len	Meaning	Reference
19	Forward on/off	1	Enable/Disable IP packet forwarding	RFC 2132 (33)
20	Source routing on/off	1	Enable/Disable IP packet forwarding for packets specifying nonlocal source routes	RFC 2132 (33)
21	Policy filter	*N*	Specifies acceptable nonlocal next hops to which IP packets may be forwarded for packets specifying nonlocal source routes	RFC 2132 (33)
22	Max datagram reassembly size	2	The maximum size datagram the client should be ready to reassemble specified as a 16-bit unsigned integer	RFC 2132 (33)
23	Default IP TTL	1	Default IP time to live value for use in outgoing packets' IP header TTL field	RFC 2132 (33)
24	Path MTU aging timeout	4	The timeout in seconds when performing path maximum transmission unit (MTU) discovery in accordance with RFC 1191; MTU discovery helps minimize packet fragmentation along the path	RFC 2132 (33)
25	Path MTU plateau table	*N*	A table listing MTU sizes to use when performing path MTU discovery per RFC 1191	RFC 2132 (33)
26	Interface MTU	2	The value of the MTU for this device interface	RFC 2132 (33)
27	All subnets are local	1	Indicates whether all subnets within the client's network use the same MTU as the local subnet to which the client is connected	RFC 2132 (33)
28	Broadcast address	4	Specifies the broadcast IP address for the client's subnet	RFC 2132 (33)
29	Mask discovery	1	Specifies whether the client should perform subnet mask discovery or not	RFC 2132 (33)
30	Mask supplier	1	Specifies whether the client should respond to other clients performing mask discovery	RFC 2132 (33)
31	Router discovery	1	Specifies whether the client should perform router discovery or not	RFC 2132 (33)
32	Router solicitation address	4	Specifies the IP address to which the client should direct router solicitation requests	RFC 2132 (33)
33	Static route	*N*	Specifies a set of static routes the client should install in its routing cache; listed as "destination network – next hop router" pairings (obsoleted by RFC 3442 defining the classless static route option, 121).	RFC 2132 (33) RFC 3442 (36)

34	Trailer encapsulation	1	Specifies whether the client should attempt to negotiate the use of layer 2 frame trailers (like headers but at the end of the frame payload) in ARP messages	RFC 2132 (33)
35	ARP timeout	4	ARP cache timeout in seconds	RFC 2132 (33)
36	Ethernet encapsulation	1	Specifies whether the client should use Ethernet II or IEEE 802.3 on an Ethernet interface	RFC 2132 (33)
37	Default TCP TTL	1	Default TCP time to live value	RFC 2132 (33)
38	TCP keepalive time	4	TCP keepalive interval in seconds	RFC 2132 (33)
39	TCP keepalive garbage	1	Specifies whether the client should send an octet of "garbage" within TCP keepalive messages for compatibility with older implementations	RFC 2132 (33)
40	NIS domain	*N*	Network Information Services (NIS) domain	RFC 2132 (33)
41	NIS servers	*N*	*N*/4 Network Information Services server addresses	RFC 2132 (33)
42	NTP servers	*N*	*N*/4 Network Time Protocol server addresses	RFC 2132 (33)
43	Vendor specific	*N*	Vendor-specific information	RFC 2132 (33)
44	NETBIOS Name server	*N*	*N*/4 NETBIOS Name server (aka WINS server) addresses	RFC 2132 (33)
45	NBDD server	*N*	*N*/4 NETBIOS Datagram Distribution (NBDD) server addresses	RFC 2132 (33)
46	NETBIOS node type	1	Specifies the client as a specific NETBIOS node type	RFC 2132 (33)
47	NETBIOS scope	*N*	Specifies the NETBIOS scope for the client	RFC 2132 (33)
48	X Window font server	*N*	*N*/4 X Window font server addresses	RFC 2132 (33)
49	X Window display manager	*N*	*N*/4 X Window display manager addresses	RFC 2132 (33)
50	Address request	4	IP address requested by the client (within a Discover message)	RFC 2132 (33)
51	Address time	4	IP address lease time requested by the client (within a Discover or Request message)	RFC 2132 (33)
52	Option overload	1	Indicates that the "sname" and/or "file" DHCP header fields contain additional DHCP option information if options to return to the client exceed the normal option space in the message	RFC 2132 (33)
53	DHCP message type	1	DHCP message type as we discussed earlier in this chapter (Discover, Offer, etc.)	RFC 2132 (33)

(*Continued*)

TABLE 4.1. DHCP Options (*Continued*)

Code	Name	Len	Meaning	Reference
54	DHCP server identifier	4	DHCP server identification provided in the Offer (and Request and optionally ACK, NAK) to identify the server, for example, to distinguish among multiple offers	RFC 2132 (33)
55	Parameter list	*N*	List of DHCP option code numbers for parameters requested by the client	RFC 2132 (33)
56	DHCP error message text	*N*	Text containing an error message; can be used by the server in a Nak message to the client or by the client in a Decline message; for example, this text could be included in logging details	RFC 2132 (33)
57	Maximum DHCP message size	2	The maximum DHCP message length the client is willing to accept	RFC 2132 (33)
58	Renewal (T1) time	4	Interval from address assignment time to the time the client enters the Renewing state	RFC 2132 (33)
59	Rebinding (T2) time	4	Interval from the address assignment time to the time the client enters the Rebinding state	RFC 2132 (33)
60	Vendor class identifier	*N*	Used by clients to specify a vendor-specific identifier	RFC 2132 (33)
61	Client Id	*N*	Client identifier	RFC 2132 (33)
62	Netware/IP domain	*N*	Netware/IP domain name	RFC 2242 (37)
63	Netware/IP option	*N*	Netware/IP sub options	RFC 2242 (37)
64	NIS + domain	*N*	NIS + client domain name	RFC 2132 (33)
65	NIS + servers	*N*	*N*/4 NIS + server addresses	RFC 2132 (33)
66	TFTP server name	*N*	TFTP server name; can be used when the "sname" DHCP header field has been overloaded with other options	RFC 2132 (33)
67	Bootfile name	*N*	Bootfile name; can be used when the "file" DHCP header field has been overloaded with other options	RFC 2132 (33)
68	Home agent	*N*	*N*/4 Mobile IP home agent addresses	RFC 2132 (33)
69	SMTP server	*N*	*N*/4 Simple Mail Transfer Protocol (SMTP) server addresses for outgoing email	RFC 2132 (33)

70	POP3 server	*N*	*N*/4 Post Office Protocol v3 (POP3) server addresses for incoming email retrieval	RFC 2132 (33)
71	NNTP server	*N*	*N*/4 Network News Transport Protocol (NNTP) server addresses	RFC 2132 (33)
72	WWW server	*N*	N/4 World Wide Web (WWW) server addresses	RFC 2132 (33)
73	Finger server	*N*	*N*/4 Finger server addresses; finger servers enable retrieval of host user information regarding login name, login duration, and more	RFC 2132 (33)
74	IRC server	*N*	*N*/4 Internet Relay Chat (IRC) server addresses	RFC 2132 (33)
75	StreetTalk server	*N*	N/4 StreetTalk server addresses; StreetTalk was a Banyan Vines user and resource directory	RFC 2132 (33)
76	STDA server	*N*	*N*/4 StreetTalk Directory Assistance (STDA) server addresses; StreetTalk was a Banyan Vines user and resource directory	RFC 2132 (33)
77	User class	*N*	User class identifier	RFC 3004 (38)
78	SLP directory agent	$N + 1$	*N*/4 Service Location Protocol (SLP) Directory Agent IP address(es)	RFC 2610 (39)
79	SLP service scope	*N*	SLP service scope the SLP agent is configured to use	RFC 2610 (39)
80	Rapid Commit	0	Rapid Commit – requests a two-packet DHCP transaction instead of the normal four-packet DORA process for mobility or overhead-constrained applications	RFC 4039 (40)
81	Client FQDN	*N*	Fully qualified domain name (FQDN) – defines the client's FQDN and whether the client or DHCP server should update DNS	RFC 4702 (41)
82	Relay agent information	*N*	Relay agent information – additional client information supplied by the intervening relay agent	RFC 3046 (42)
83	Internet storage name service (iSNS)	*N*	iSNS server addresses and iSNS application information	RFC 4174 (43)
84	Unassigned	–	–	RFC 3679 (44)
85	NDS servers	*N*	N/4 Novell Directory Services (NDS) server IP addresses to contact for NDS client authentication and access the NDS directory repository	RFC 2241 (45)
86	NDS tree name	*N*	NDS tree name of the NDS repository the client should contact	RFC 2241 (45)
87	NDS context	*N*	NDS initial context within the NDS repository the NDS client should use	RFC 2241 (45)
88	Broadcast and multicast server (BCMCS) controller domain name	*N*	BCMCS domain name (FQDN) list, used to construct follow-up SRV query(ies) (BCMCS is used in 3G wireless networks to enable mobiles to receive broadcast and multicast services)	RFC 4280 (46)

(*Continued*)

TABLE 4.1. DHCP Options (*Continued*)

Code	Name	Len	Meaning	Reference
89	BCMCS Controller IPv4 address	*N*	*N*/4 BCMCS Controller IP address(es) (BCMCS is used in 3G wireless networks to enable mobiles to receive broadcast and multicast services)	RFC 4280 (46)
90	Authentication	*N*	Authentication option used to communicate authentication information between the client and the server in accordance with the DHCP authentication protocol	RFC3118 (47)
91	Client last-transaction time option	4	Seconds since the last DHCP transaction with the client on this lease as queried in a DHCP Lease Query message	RFC 4388 (48)
92	Associated IP option	*N*	List of IP addresses associated with the client as queried in a DHCP Lease Query message	RFC 4388 (48)
93	PXE client system	*N*	PXE client system architecture type(s) each encoded as 16-bit code, for example, Intel x86PC, DEC Alpha, EFI x86-64, and so on	RFC 4578 (49)
94	PXE client network interface	3	PXE client network interface identifier with individual octets encoded for interface type, interface major version number, and interface minor version number	RFC 4578 (49)
95	LDAP	*N*	Lightweight Directory Access Protocol servers; this option is used by Apple Computer though no governing RFC has been published	RFC 3679 (44)
96	Unassigned	–	–	RFC 3679 (44)
97	PXE client machine identifier	*N*	PXE client machine identifier with encoded type and identifier value	RFC 4578 (49)
98	User Authentication Protocol (UAP)	*N*	List of locations (URLs) for services capable of processing authentication requests encapsulated using Open Group's UAP	RFC 2485 (50)
99	Civic location	–	Location of the server, network element closest to the client or the client itself as provided by the server encoded in country-specific civic (e.g., postal) format	RFC 4776 (51)
100	Time zone	*N*	Time zone encoded as IEEE 1003.1 TZ (POSIX)	RFC 4833 (52)
101	Time zone database	*N*	Reference to a local (on the client) TZ database for lookup of time zone	RFC 4833 (52)
102–111	Unassigned	–	–	RFC 3679 (44)

112	Netinfo address	*N*	NetInfo parent server address: this option is used by Apple Computer though no governing RFC has been published; NetInfo is a distributed database user and resource information for Apple devices	RFC 3679 (44)
113	Netinfo tag	*N*	NetInfo parent server tag: this option is used by Apple Computer though no governing RFC has been published. NetInfo is a distributed database user and resource information for Apple devices	RFC 3679 (44)
114	URL	*N*	Uniform resource locator; this option is used by Apple Computer though no governing RFC has been published	RFC 3679 (44)
115	Unassigned	–	–	RFC 3679 (44)
116	Autoconfigure	1	Instructs the client to autoconfigure a link local address (69.254.0.0/16) or not. This can be used by the DHCP server to inform the client that it has no IP addresses to assign and that the client may or may not autoconfigure	RFC 2563 (53)
117	Name service search	*N*	Lists one or more name services in priority order that the client should use for name resolution: DNS, NIS, NIS+, or WINS	RFC 2937 (54)
118	Subnet selection	4	Identifies an IP subnet (address) from which to allocate an IP address to this client – overrides the GIAddr setting or DHCP server interface on which a broadcast Discover was received	RFC 3011 (55)
119	Domain search	*N*	List of one or more domains for configuration of the client's resolver. If the application requests a resolution for a non-FQDN host name, these domain(s) will successively be appended to the host name prior to querying	RFC 3397 (56)
120	SIP servers	*N*	A listing of one or more of Session Initiation Protocol (SIP) server FQDN (s) or of SIP server IP address(es). SIP is a control protocol for management of multimedia calls or sessions	RFC 3361 (57)
121	Classless static route	*N*	Specifies a set of static routes the client should install in its routing cache; listed as "<CIDR mask length>.<destination network> – next hop router" pairings. The destination network is enumerated only to significant octets, dropping local (nonsubnet) portions; for example, 172.16.0.0/12 would be encoded as 12.172.16 and 10.0.0.0/18 as 18.10.0.0	RFC 3442 (58)

(*Continued*)

TABLE 4.1. DHCP Options (*Continued*)

Code	Name	Len	Meaning	Reference
122	CableLabs client configuration	*N*	Specifies resource (e.g., provisioning server, DHCP server, etc.) locations, and parameters for use by cable multimedia terminal adapters (MTAs), which are customer premises devices operating over a DOCSIS cable network, providing VoIP and related multimedia services	RFC 3495 (59)
123	Location configuration information (LCI)	16	Provides the client its LCI, including latitude, longitude, altitude, and resolution of each coordinate	RFC 3825 (60)
124	Vendor-identifying vendor class	*N*	Enables specification of multiple vendor classes, each identified by IANA-assigned enterprise number (EN); this is useful to identify the hardware vendor, software vendor, application vendor, and so on supporting the device	RFC 3925 (61)
125	Vendor-identifying vendor-specific information	*N*	Set of DHCP options grouped by vendor as identified by IANA-assigned EN	RFC 3925 (61)
126–127	Unassigned	–	–	RFC 3679 (44)
128	PXE – undefined (vendor specific)			RFC 4578 (49)
Overloaded	Etherboot signature. 6 bytes: E4:45:74:68:00:00			
	DOCSIS “full security” server IP address			
	TFTP server IP address (for IP phone software load)			
129	PXE – undefined (vendor specific)			RFC 4578 (49)
Overloaded	Kernel options. Variable length string			
	Call server IP address			
130	PXE – undefined (vendor specific)			RFC 4578 (49)
Overloaded	Ethernet interface. Variable length string			
	Discrimination string (to identify vendor)			
131	PXE – undefined (vendor specific)			RFC 4578 (49)

Overloaded	Remote statistics server IP address			
132	PXE – undefined (vendor specific)			RFC 4578 (49)
Overloaded	802.1Q VLAN ID			
133	PXE – undefined (vendor specific)			RFC 4578 (49)
Overloaded	802.1D/p L2 priority			
134	PXE – undefined (vendor specific)			RFC 4578 (49)
Overloaded	Diffserv code point			
135	PXE – undefined (vendor specific)			RFC 4578 (49)
Overloaded	HTTP proxy for phone-specific applications			
136	PANA agent	*N*	Identifies one or more IPv4 addresses of PANA (Protocol for Carrying Authentication for Network Access) authentication agents for use by the client for authentication and authorization for network access service	RFC 5192 (62)
137	LoST server	*N*	Location to service translation (LoST) server domain name; LoST protocol maps service identifiers and location information to service URLs	RFC 5223 (63)
138	CAPWAP access controller	*N*	Control and provisioning of wireless access points (CAPWAPs) access controller IP address(es) to which the client may connect	RFC 5417 (64)
139	Mobility service (MoS) IP addresses	*N*	IPv4 address(es) for servers providing particular types of IEEE 802.21 MoS	RFC 5678 (65)
140	MoS FQDNs	*N*	FQDN(s) for servers providing particular types of IEEE 802.21 MoS	RFC 5678 (65)
141	SIP user agent configuration		Session Initiation Protocol (SIP) user agent configuration	draft-lawrence-sipforum-user-agent-config-03.txt (179)
142–149	Unassigned			RFC 3942 (66)
150	TFTP server address			RFC 5859 (175)
	Etherboot			
	GRUB configuration path name			

(*Continued*)

TABLE 4.1. DHCP Options (*Continued*)

Code	Name	Len	Meaning	Reference
151–174	Unassigned			RFC 3942 (66)
175	Etherboot (tentatively assigned – June 23, 2005)			
176	IP telephone (tentatively assigned – June 23, 2005)			
177	Etherboot (tentatively assigned – June23, 2005)			
178–207	Unassigned			RFC 3942 (66)
208	PXE magic (deprecated)	4	F1:00:74:7E	RFC 5071 (67)
209	PXE configuration file	*N*	Configuration file name or file path name for second-stage PXE boot loading	RFC 5071 (67)
210	PXE path prefix	*N*	Configuration file path prefix to the file name specified in the PXE configuration file option (209)	RFC 5071 (67)
211	PXE reboot time	4	Number of seconds to wait to reboot if TFTP server is unreachable	RFC 5071 (67)
212	6rd configuration		6rd customer edge device configuration (6rd is a service provider IPv4-IPv6 technology - see Chapter 15)	RFC 5969 (176)
213	LIS domain name		Location Information Server (LIS) domain name for this access network	draft-ietf-geopriv-lis-discovery-15.txt (178)
214–219	Unassigned			
220	Subnet allocation option (tentatively assigned – June 23, 2005)			
221	Virtual subnet selection option (tentatively assigned – June 23, 2005)			
222–223	Unassigned			RFC 3942 (66)
224–254	Reserved (private use)			
255	End	0	None	RFC 2132 (33)

*The *N*/4 notation refers to the use of "*N*" bytes to represent one or more IPv4 addresses, each of which is comprised of four bytes; thus for a length of *N*, the field would contain *N*/4 complete IPv4 addresses. This implies of course that *N* is a multiple of 4 in cases where the data type is IP address.

†IEN-116 = Internet Experiment Note 116; IENs were eventually merged with RFCs as TCP/IP went into production across ARPANET.

option 172. We've already described how client classes can be used to configure the DHCP server to distinguish different DHCP clients. We can now associate options for each pool, which will be provided to clients receiving addresses in the corresponding pool. An example of this is depicted in the high-level sample configuration in Figure 4.5b, which includes option declarations with the class and address pool statements to define additional parameters to be provided to clients. Alternatively, Manual DHCP address reservations enable mapping of a hardware address to a specific IP address, and associated DHCP options can also be defined for the device.

Table 4.1 lists the current set of defined DHCP options. The "code" column indicates the option code or number and the "name" column lists the corresponding option name. Note that the "Len" (length) column indicates the value of the length field within the option. The total option length is this value plus two bytes, one byte for the code and one for the length field itself.

4.5 OTHER MEANS OF DYNAMIC ADDRESS ASSIGNMENT

While DHCP provides a means for network administrators to preallocate dynamic address pools on a number of subnets and provide a mechanism to discern different device types for discriminatory assignment of an IP address and configuration parameters, there are other methods, albeit less popular, for dynamic address assignment. A common alternative method besides address autoconfiguration is the use of a Radius server to assign an IP address. Radius, or its successor protocol, Diameter, provides an authentication, authorization, and accounting (AAA) service for IP hosts attempting to access a network. The connection from a client to a Radius server is commonly performed via a Point-to-Point (PPP) or Extensible Authentication Protocol (EAP) connection, for example, when the client is attempting to access a network edge device or dial pool. The Radius server challenges the client to enter a user name and password, authenticates the entered information against its internal or external database, and finally provides access to the network by providing an IP address to the client.

While vastly simplifying the Radius protocol, the relevant concept here is that some Radius servers, or even edge router devices, can be configured with address pools from which individual IP address assignments can be made to authorized clients. In some cases, Radius servers can be configured to actually utilize the DHCP protocol to obtain an address from a DHCP server. In this case, the Radius server acts as a DHCP proxy client to obtain an IP address on behalf of, and for assignment to, the requesting client. We'll discuss some alternative DHCP server deployment strategies in Chapter 7, where we'll compare DHCP deployment on edge devices with DHCP deployment on discrete DHCP servers.