

IOS DHCP Services Part 1

Michael J. Martin

One of the things about writing a series of technical articles is that you need to tackle topics that are obvious to some, but new to others. The trick is to hopefully strike a balance between providing a valuable foundation of information to those new to the topic and providing something new to the experts so that no one's time is wasted.

This month's article is the first of two-part series on the Dynamic Host Configuration Protocol (DHCP) and Cisco IOS support for DHCP services. In this article we will examine what DHCP is and how it works. We will look DHCP at the key transaction and packet level components in order to understand its overall operation so that we may implement and troubleshoot it effectively. This will not be a bits and bytes overview, rather we will examine DHCP from a practical hands-on viewpoint, providing administrators with the information they need to support and implement DHCP services in an IOS environment.

What is DHCP?

One of the more appealing aspects of IPX and AppleTalk is the dynamic host address assignment capability. The Dynamic Host Configuration Protocol (DHCP) is a client/server framework for dynamically providing IP address and supporting configuration information for hosts on IP networks. According to the RFC, three methods of allocation are available: Automatic, for assigning a permanent address to a host; Dynamic, for assigning a temporary address for a finite period; and Manual, in which an administrator assigned address is conveyed to the host. While dynamic IP address assignment is by far DHCP's most common use, when implemented properly DHCP is a powerful mechanism for managing access and configuration policy. That makes it an ideal tool, not only for managing your desktop, but your servers as well.

DHCP is built upon the Bootstrap Protocol (BOOTP) defined in RFC 951 as a mechanism for downloading address and boot configuration for diskless workstations. There are two major differences between the two protocols. DHCP provides the capability to assign fixed leases. DHCP can provide an IP address, plus a number of other configuration attributes or configuration attributes to hosts with static IP addresses. BOOTP only provides the host the IP address. DHCP/BOOTP has three components. The server is a host with a static IP address that allocates, distributes, and manages IP and configuration data assignments. Each allocation (IP and config data) is stored on the server in a data set called a binding. The client is any device using DHCP as a method for obtaining IP addressing or supporting configuration information. The relay agent is a gateway device (typically a router) that functions as an intermediary between a client on one subnet and a server on another.

A typical DHCP implementation utilizes one or two (one primary and one secondary) servers and (depending on network size) a number of relay agents. When DHCP was first showing up on networks, many administrators implemented one server per subnet (this was quite common in Microsoft environments where the server also functioned as the WINS server and Master Browser) or loaded a server with multiple NICs and directly attached the server multiple subnets. While this approach works, it is quite costly and clearly has some scaling issues. Today, a good rule of thumb is to have a primary and backup server per physical location. While it is possible to forward DHCP requests over WAN/VPN links using a relay, this not really effective (unless you're using DHCP to address remote links -- more on this later).

DHCP Messages

Client/server data exchange is done using messages, some of which are sent only by the client, some only by the server. When a client has requested and accepted an IP address, the interface is in what the RFC refers to as a bound state. (This is why the allocation data sets are called bindings.) The IP address allocation itself is referred to as a lease. Transport of the messages is accomplished with UDP as the transport protocol using BOOTP's client and server port definitions. Client

IOS DHCP Services Part 1

Michael J. Martin

messages originate from port UDP 68 (bootpc/dhpc) with a destination of UDP port 67 (bootps/dhcps). Conversely, server messages originate from UDP port 67 with a destination of UDP port 68. DHCP messages follow the same format as BOOTP messages. (The VEND field in the BOOTP message spec is the OPTION field in the DHCP spec.) While it is technically a separate protocol standard, DHCP is an extension of the BOOTP protocol. This "dual-use" aspect of DHCP and BOOTP requires DHCP to be backward compatible with the BOOTP specification to support BOOTP services and respond when appropriate with BOOTP messages. There are two BOOTP message types:

- BOOTREQUEST: Sent as an L2/L3 broadcast by the client to solicit the server for IP and configuration data.
- BOOTREPLY: Sent as a unicast to the client with configuration data.

The original DHCP specification, RFC 1541 defined seven DHCP message types. The draft standard RFC 2131 defines eight message types:

- DHCPDISCOVER: Sent as a L2/L3 broadcast by DHCP clients, used to discover and solicit a lease.
- DHCPOFFER: Sent as a L2/L3 unicast by DHCP servers, used to make an initial lease offer to a client.
- DHCPREQUEST: Sent as a broadcast or unicast by DHCP clients to a server accepting and/or verifying a DHCP lease. Also used to extend a DHCP lease.
- DHCPDECLINE: Sent as a L2/L3 broadcast by a client to a server to inform the server the address is in use.
- DHCPACK: Sent as a unicast by the server to the client to transmit and confirm lease information.
- DHCPNAK: Sent as a unicast by the server to inform the client the lease information they have is invalid.
- DHCPRELEASE: Sent as unicast by the client to inform the server it is releasing the address.
- DHCPINFORM: Sent as unicast, used by the client to request more configuration data from the server (the client has a fixed IP and requires gateway, DNS, etc. support information).
- DHCP messages are typically no longer than 576 bytes. (They can be longer at the client's option.)

The DHCP messages have fixed fields and optional fields. The fixed fields have a specific binary length and are present in each message. They do not always contain values of a functional nature, but can contain placeholder data to allow the message to conform to the required size. The key fixed message fields are:

- OP: The "Option Code" field indicates the type of message being sent used in client and server generated messages. It is a compatibility carryover from BOOTP. Value will indicate the message type as a BOOTREQUEST or a BOOTREPLY. (This should not be confused with DHCP options, which we will discuss in a moment.)

IOS DHCP Services Part 1

Michael J. Martin

- **XID:** Used by the clients to identify server responses.
- **CIADDR:** The "Client IP Address" is set by the client only when a host is in a bound state, i.e., when the IP address has been confirmed by the client and is in use.
- **YIADDR:** "Your IP address" is set only in server-generated messages to inform the client of their IP address.
- **GIADDR:** If the message is forwarded by a DHCP relay, the IP address of the relay is set by the relay using this field.
- **CHADDR:** The client's L2 address, set by the client and used only in client generated messages.
- **OPTIONS:** There are a number of optional configuration values that an administrator can distribute using the DHCP server. These values are set by the server and sent to the client in DHCP OFFER and DHCP ACK messages and sent back to the server in client DHCP REQUEST, DHCP DECLINE, and DHCP RELEASE messages.

There are over 100 DHCP client configuration options (of a potential 255). The standard DHCP message can support up to 312 bytes of option values. Each DHCP option is a variable length data set, so the number of options that can be transmitted in a single message is dependent on the options being set. Most DHCP options are structured using the three-part format Option Code: Option Message Length: Option Data. Not all options are valid for every environment. The more commonly used "generic" options are listed here. (The number in parenthesis is the Option Code followed by the TCPDump Translation.)

- **Client Identifier (61/CID):** The CID is used by the server to distinguish clients. It is set on the client and sent along in all of the client-generated messages. The CID is entered as a text string, but sent as a hex string. While the CID is an alternative to using the client's L2 Media Access Control (MAC) address contained in the CHADDR field, many DHCP implementations that support the CID option will send a random hex string if none is supplied. This is the preferred method for associating a host with a static address lease.
- **Server Identifier (54/SID):** The SID is used by the client to determine the origin of a DHCP message and in DHCP REQUEST messages to indicate the recipient server.
- **Lease Time (51/LT):** The duration of time the lease is active for. This can be a finite time span expressed in seconds or set for infinite (indicated by a FFFFFFFF) in the data field.
- **Lease Renewal Time (58/RN):** The time that the host begins to extend its lease via a unicast DHCP REQUEST.
- **Lease Renewal Time -- rebinding (59/RB):** The time that the host begins to find a new server to extend its lease via a unicast DHCP REQUEST.
- **Subnet Mask (1/SM):** Used by the server to set the client's subnet mask
- **Parameter Request List (55/PR):** A list of configuration option values wanted by the client.
- **Broadcast Address (28/BA):** Used by the server to set the host's broadcast address.

IOS DHCP Services Part 1

Michael J. Martin

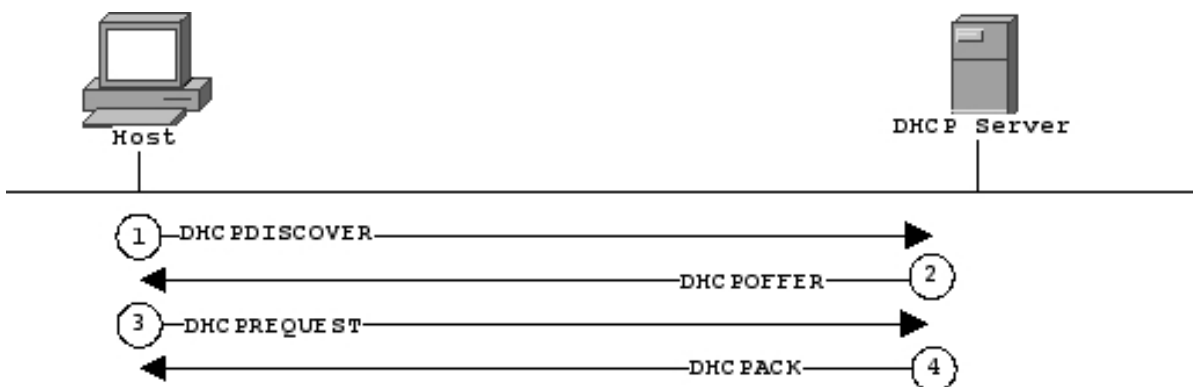
- Host Name (12/HH): Used by the server to provide a client's a hostname.
- Domain Name (15/DN): Used by the server to assign a client's domain name.
- DNS-Server (6/NS): Sets DNS servers, up to four.
- Static Routes (33/SR): Sets static routes to be added to the client's route table. Up to four route entries are formatted as two 32-bit integers.
- Time Server (4/NTP): Sets time servers the clients should sync to, up to four.
- Time Offset (2): Sets the client's time zone in seconds. For example EST is -5 from zero meridian, so the value is -18,000.
- Routers (3/DG): Used by the server to add (up to four) default gateways to the client's routing table.
- Interface MTU (26/MTU): Sets the client Maximum Transmission Unit.
- NetBIOS Node Type (46/WNT): Sets the NetBIOS node type (determines NetBIOS name resolution method) B-Node = Broadcast Node (NetBIOS resolution done using broadcasts only), P-Node = Peer Node (WINS Only name resolution) M-Node = Mixed Node (broadcast, then WINS), H-Node = Hybrid Node (WINS, broadcast).
- NetBIOS Name Server (45/WNS): Sets WINS servers, up to four.
- NetBIOS Scope ID (47/WSC): Sets the NetBIOS scope.
- Vendor Class Identifier (60/VC): Informs the server about the type of host requesting configuration data. Used by the server to interpret the vendor-specific information (43/VO) option.

DHCP Client/Server Interactions

All DHCP processes are client driven. There are three basic DHCP client/server operations: obtaining a lease, renewing a lease and terminating a lease.

Obtaining A Lease

At boot time (startup and reboot and post-release of a DHCP lease), the DHCP client starts off in the INIT-REBOOT state. "INIT" implies that they are first joining the network and have no previous IP address lease. "REBOOT" infers that the host has a previously assigned lease that it wishes to use and needs validated by the DHCP server. Obtaining a lease moves from the INIT-REBOOT state to the BOUND state through a four-step process.



IOS DHCP Services Part 1

Michael J. Martin

The lease process is initiated by broadcasting a DHCPDISCOVER message across the IP subnet. The DHCPDISCOVER message is used to find the DHCP servers on the network. The server can be either on the host's local subnet or on a remote subnet. When the server is on a remote subnet, the router typically functions as a DHCP relay server and forwards the DHCP broadcast requests (as unicast) to the DHCP server. Because the host has no valid IP information at boot, it uses L2 and L3 broadcast addresses to communicate with the server. The L2 header has the client's MAC address as the frame's source address and the MAC broadcast address FF:FF:FF:FF:FF:FF as the destination. The XID is generated by the client and used to identify server responses. Here is an L3 packet dump of the DHCPDISCOVER message: 0.0.0.0.bootpc > 255.255.255.255.bootps: xid:0xd467362f DHCP:DISCOVER CID:01:00:40:96:39:f5:04 HN:"brio" VC:77.83.70.84.32.53.46.48 PR:SM+DN+DG+NS+WNS+WNT+WSC+RD+SR+VO

The L3 packet dumps were collected using the following command: tcpdump -i {interface} -s 0 -vvv dst port {tcp/udp port number}.

A lot of information about the host can be determined by looking at the DISCOVER request. First, there is the Client ID. While it's in hex it can be translated easily using any hex editor. It is often host generated, but many users will enter either their name or a hostname. This host has its hostname statically configured. For binding consistency any statically configured variable will be transmitted by the client to the server. It also allows the server flexibility in how it decides to associate static bindings with specific client hosts. One last tidbit about the DHCPDISCOVER message -- it is also possible to identify the type host requesting an address by looking at the Parameter Request List (PR). Notice in the example the options WNS+WNT+WSC. These options request NetBIOS configuration options, so it's a good bet this is a Windows-based system. Since the DHCPDISCOVER message is sent as a broadcast, it is blind to the receiver. So one or more servers can receive the message, resulting in the client receiving multiple lease offers. The client will only accept one of the offers (usually, the first lease offer received) and will inform all of the servers of its selection. While the number of Option fields may vary, at a minimum a DHCPDISCOVER message will contain the following values:

Fixed fields:

- OP
- XID
- CHADDR (may not have an entry if the CID is supported by the client)
- Optional fields:
 - Message Type = DHCPDISCOVER
 - Vendor Class Identifier
 - Parameter Request List
 - Any statically defined Option values

When a DHCP server receives a DHCPDISCOVER message, it examines the message for values it may use for static allocations, determines a lease address, creates an ARP entry for the lease, and transmits a binding offer with a DHCPOFFER message. The DHCPOFFER message is sent as a unicast, using the server's L2 MAC address as the source address and the L2 address of the client (or local facing relay). Here is look at the L3/L4 values of the DHCPOFFER:

```
172.30.71.1.bootps > 172.30.71.8.bootpc: xid:0xd467362f Y:172.30.71.8
DHCP:OFFER SID:172.30.71.1 LT:432000 RN:216000 RB:378000 SM:255.255.255.0
NS:192.168.100.45 DG:172.30.71.1 DN:"core.outland.net" WNT:8
```

IOS DHCP Services Part 1

Michael J. Martin

The server addresses the L3 packet with its L3 address as the source, using the candidate lease address as the destination. At this time the client has no valid L3 address, so delivery of the message is accomplished using the L2 address of the client. If a DHCP relay is involved, the server will forward the message as a unicast to the relay using the relay's MAC address for the candidate lease IP address. The relay will process the message and create its own ARP entry for the candidate lease and forward the message on to the client via unicast. The message itself contains the candidate lease in the YIADDR (Y) fixed message field and the server identifies itself using the Server ID (SID) option and the XID fixed field with the same value as the clients DHCPDISCOVER message. A properly formatted DHCPOFFER will contain these key values:

- Fixed fields:
- Transaction ID (from the DHCPDISCOVER message)
- Your Client Address (the server-assigned client IP address)
- Gateway Address (if the message has been handled by a DHCP relay)
- Client Hardware Address
- Options:
- Message Type = DHCPOFFER
- Server Identifier
- Address Lease Time
- Address Lease Renew Time
- Address Lease Rebind Time
- Subnet Mask

Once the HOST receives the DHCPOFFER from the server, it responds back with a DHCPREQUEST message. This message is used for both lease origination and lease renewal and verification. When used as part of the lease origination process, technically the client's DHCPREQUEST is requesting the IP information be verified just after it has been assigned. The message serves to provide error checking (to make sure that the assignment is still valid, since in theory it could have been offered to another host during this process) and serves as a binding acceptance notice to the selected server and an implicit decline to any other servers (via the Server Identifier Field) that may have provided the host a binding offer. The DHCPREQUEST message is also sent as a broadcast. The L2 source is the host's, and the L2 destination is the L2 broadcast. Here is the L3 packet dump:

```
0.0.0.0.bootpc > 255.255.255.255.bootps: xid:0xd467362f DHCP:REQUEST
CID:01:00:40:96:39:f5:04 RQ:172.30.71.8 SID:172.30.71.1
VC:77.83.70.84.32.53.46.48 PR:SM+DN+DG+NS+WNS+WNT+WSC+RD+SR+VO
```

The client uses the low-network broadcast address as the source address and the high-network broadcast as the destination. The same XID has been generated and sent along with the target server (SID) and requested lease address (RQ). Like the DHCPBROADCAST message, the Option fields will vary depending on the client configuration. The DHCPREQUEST required data points are:
Fixed Fields:

Transaction ID (new)

Options:

Server Identifier

Message Type = DHCPREQUEST

Requested IP address (The Server Assigned, Client IP Address)

Vendor Class Identifier

Parameter Request List

Any statically defined OPTION values

IOS DHCP Services Part 1

Michael J. Martin

Upon receiving the DHCPREQUEST message, the server verifies the lease information, creates a new ARP entry for the client lease and replies with a unicast DHCPACK message.

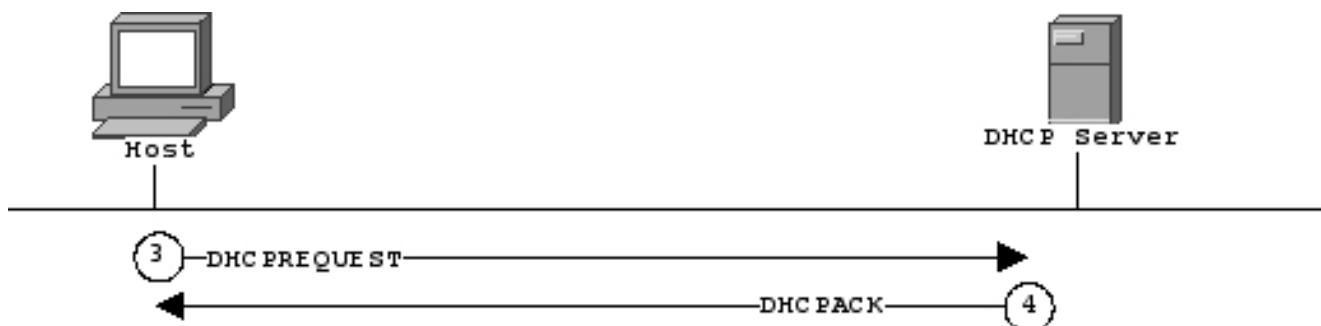
```
172.30.71.1.bootps > 172.30.71.8.bootpc: xid:0xd467362f Y:172.30.71.8 DHCP:ACK  
SID:172.30.71.1 LT:432000 RN:216000 RB:378000 SM:255.255.255.0 NS:204.60.0.3  
DG:172.30.71.1 DN:"core.outland.net" WNT:8
```

Notice that it's almost an exact duplicate of the DHCP OFFER, except for the change in the Message Type Option.

When the host receives the DHCPACK message, it logs the configuration information and performs an ARP lookup for the assigned address. If no reply is found, the configuration data is used to configure the host's TCP/IP stack.

Renewing And Verifying A Lease

Hosts crash, leave the network (laptops do go home) and sometimes need to be rebooted. Once a host has a lease, it needs to re-verify the information whenever the interface is reset. The lease time Option informs the host how long it can keep its lease. Once the server has issued a lease, the IP address is removed from the allocation pool. As long as the lease time is still valid, each time the host joins the network it has to validate its address using a DHCPREQUEST sent as a broadcast. If the address information is still valid, the server responds with a DHCPACK. If the address has been reallocated (because the host has not renewed the lease within the subscribed time window) the server will respond with a DHCPNACK and the host will revert to an INIT state and request a new binding.



A similar process is followed for renewing a lease. A dynamic lease assignment is technically finite. Most administrators adjust their lease time according to how the network is used. For example, if the subnet has a lot of churn, with hosts coming and going with a great amount of frequency, the lease time should be short (say an hour or so) to make sure the address pool is being recycled and that addresses will always be available. If the network is mostly user desktops or servers (with static IPs using DHCP for gateway, name server and other configuration data) then the leases can be set for longer. Any changes in the network configuration, like a change in the name server's address or a new additional default gateway can be syndicated with some regularity.

When a client's lease is close to expiring (determined with the Lease Renewal Option value), the client sends a DHCPREQUEST requesting a lease time extension. If the extension is acceptable to the server, it responds with DHCPACK and a new lease time. If there is a conflict, the server responds with a DHCPNACK and the client reverts to an INIT state and requests a new lease.

IOS DHCP Services Part 1

Michael J. Martin

Releasing a Lease

For administrative purposes, a host may want to release its lease. This is accomplished using a DHCPRELEASE message. While not a requirement, some clients issue a release when the host shuts down. Here is the client message:

```
08:40:30.679389 172.30.71.8.bootpc > 172.30.71.1.bootps xid:0xc07b0d17  
C:172.30.71.8 DHCP:RELEASE SID:172.30.71.1 CID:01:00:40:96:39:f5:04
```

This message is sent as a unicast from the client to the server. Notice that the CIADDR fixed field is used. Below is the server response. First an DHCPOFFER is sent, followed by a DHCPACK (the timestamps have been left so you can see the message order in real time):

```
08:38:33.435562 172.30.71.1.bootps > 172.30.71.8.bootpc xid:0xd467362f  
Y:172.30.71.8 DHCP:OFFER SID:172.30.71.1 LT:432000 RN:216000 RB:378000  
SM:255.255.255.0 NS:204.60.0.3 DG:172.30.71.1 DN:"core.outland.net" WNT:8  
08:38:33.445398 172.30.71.1.bootps > 172.30.71.8.bootpc xid:0xd467362f  
Y:172.30.71.8 DHCP:ACK SID:172.30.71.1 LT:432000 RN:216000 RB:378000  
SM:255.255.255.0 NS:204.60.0.3 DG:172.30.71.1 DN:"core.outland.net"
```

Once the release is issued, the server is free to reassign the address to another host. When the host returns to the network, it starts form an INIT state.

By this time you should be ready to get your hands dirty implementing DHCP using the IOS, or ready for a nap. Either way, we can accommodate. Next month we will tackle approaches to implementing DHCP in a sane and secure manner, along with a theory and practice overview of implementing DHCP services on IOS-based hardware. As for today, go take that nap.