WIKIPEDIA

Dynamic Host Configuration Protocol

The **Dynamic Host Configuration Protocol** (**DHCP**) is a network management protocol used on UDP/IP networks whereby a DHCP server dynamically assigns an IP address and other network configuration parameters to each device on a network so they can communicate with other IP networks.[1] A DHCP server enables computers to request IP addresses and networking parameters automatically from the Internet service provider (ISP), reducing the need for a network administrator or a user to manually assign IP addresses to all network devices.[1] In the absence of a DHCP server, a computer or other device on the network needs to be manually assigned an IP address, or to assign itself an APIPA address, which will not enable it to communicate outside its local subnet.

DHCP can be implemented on networks ranging in size from home networks to large campus networks and regional Internet service provider networks.[2] A router or a residential gateway can be enabled to act as a DHCP server. Most residential network routers receive a globally unique IP address within the ISP network. Within a local network, a DHCP server assigns a local IP address to each device connected to the network.

Contents

History
- DHCP versions

Overview

Operation
- DHCP discovery
- DHCP offer
- DHCP request
- DHCP acknowledgement
- DHCP information
- DHCP releasing

Client configuration parameters

DHCP options
- Documented in RFC 2132
  - DHCP client vendor identification
- Documented elsewhere
  - Relay agent information sub-options

DHCP relaying

Reliability

Security

IETF standards documents

See also

Notes

References

History

In 1984, the Reverse Address Resolution Protocol (RARP), defined in RFC 903, was introduced to allow simple devices such as diskless workstations to dynamically obtain a suitable IP address. However, because it acted at the data link layer it made implementation difficult on many server platforms, and also required that a server be present on each individual network link. RARP was superseded by the Bootstrap Protocol (BOOTP) defined in RFC 951 in September 1985. This introduced the concept of a *relay agent*, which allowed the forwarding of BOOTP packets across networks, allowing one central BOOTP server to serve hosts on many IP subnets.[3]

DHCP is based on BOOTP but can dynamically allocate IP addresses from a pool and reclaim them when they are no longer in use. It can also be used to deliver a wide range of extra configuration parameters to IP clients, including platform-specific parameters.[4] DHCP was first defined in RFC 1531 in October 1993; but due to errors in the editorial process was almost immediately reissued as RFC 1541.

Four years later the DHCPINFORM message type[5] and other small changes were added by RFC 2131; which as of 2014 remains the standard for IPv4 networks.

DHCPv6 was initially described by RFC 3315 in 2003, but this has been updated by many subsequent RFCs.[6] RFC 3633 added a DHCPv6 mechanism for prefix delegation, and stateless address autoconfiguration was added by RFC 3736.

DHCP versions

In the case of DHCP, there are several RFC standards, and the version numbers individually such as DHCP v2 and v3 are not specified differently from SNMP v1, v2 and v3 in the RFC standard.[7] However, the ISC (Internet Software Consortium) DHCP server that functions as a common DHCP server has version V1, V2, and V3 in the form release 1, release 2 and release 3[8]

- ISC DHCP Release 1 (version 1): It is a more stable version with a bug fix over the initial release(beta) ISC DHCP server after 2 years.
- ISC DHCP Release 2 (version 2): DHCP Release 2 is composed of DHCP server, DHCP client and DHCP relay agent. Features include support for ping before IP allocation, and more effective prevention of errors in DHCP authentication
- ISC DHCP Release 3 (version 3): The key features of Release 3 include the ability to selectively initialize DHCP server functionality and to bundle clients that are assigned IPs into desired configuration units. In addition, since the relay agent information is selectively available, it is possible to separately manage the POOLs that are the groups to which the IP addresses are assigned, by the necessary group units. It enables the ability to leverage dynamic DNS information and DHCP authentication.
- The standard for IPv6 DHCP is defined by RFC, and the latest version is defined by DHCPv6 - RFC 3736, which includes IPv6 address Stateless autoconfiguration.

Overview

UDP/IP defines how devices on one network communicate with devices on another network, and the DHCP server can manage UDP/IP settings for devices on a network, by automatically or dynamically assigning IP addresses to the devices.

The DHCP operates based on the client–server model. When a computer or other device connects to a network, the DHCP client software sends a DHCP broadcast query requesting the necessary information. Any DHCP server on the network may service the request. The DHCP server manages a pool of IP addresses and information about client configuration parameters such as default gateway, domain name, the name servers, and time servers. On receiving a DHCP request, the DHCP server may respond with specific information for each client, as previously configured by an administrator, or with a specific address and any other information valid for the entire network and for the time

period for which the allocation (*lease*) is valid. A DHCP client typically queries for this information immediately after booting, and periodically thereafter before the expiration of the information. When a DHCP client refreshes an assignment, it initially requests the same parameter values, but the DHCP server may assign a new address based on the assignment policies set by administrators.

On large networks that consist of multiple links, a single DHCP server may service the entire network when aided by DHCP relay agents located on the interconnecting routers. Such agents relay messages between DHCP clients and DHCP servers located on different subnets.

Depending on implementation, the DHCP server may have three methods of allocating IP addresses:

Dynamic allocation
: A network administrator reserves a range of IP addresses for DHCP, and each DHCP client on the LAN is configured to request an IP address from the DHCP server during network initialization. The request-and-grant process uses a lease concept with a controllable time period, allowing the DHCP server to reclaim and then reallocate IP addresses that are not renewed.

Automatic allocation
: The DHCP server permanently assigns an IP address to a requesting client from the range defined by the administrator. This is like dynamic allocation, but the DHCP server keeps a table of past IP address assignments, so that it can preferentially assign to a client the same IP address that the client previously had.

Manual allocation (commonly called static allocation)
: The DHCP server issues a private IP address dependent upon each client's *client id* (or, traditionally, the client MAC address), based on a predefined mapping by the administrator. This feature is variously called *static DHCP assignment* by DD-WRT, *fixed-address* by the dhcpd documentation, *address reservation* by Netgear, *DHCP reservation* or *static DHCP* by Cisco and Linksys, and *IP address reservation* or *MAC/IP address binding* by various other router manufacturers. If no match for the client's *client ID* (if provided) or MAC address (if no client id is provided) is found, the server may or may not optionally fall back to either Dynamic or Automatic allocation.

DHCP is used for Internet Protocol version 4 (IPv4) and IPv6. While both versions serve the same purpose, the details of the protocol for IPv4 and IPv6 differ sufficiently that they may be considered separate protocols.[9] For the IPv6 operation, devices may alternatively use stateless address autoconfiguration. IPv6 hosts may also use link-local addressing to achieve operations restricted to the local network link.

Operation

The DHCP employs a connectionless service model, using the User Datagram Protocol (UDP). It is implemented with two UDP port numbers for its operations which are the same as for the bootstrap protocol (BOOTP). UDP port number 67 is the destination port of a server, and UDP port number 68 is used by the client.

DHCP operations fall into four phases: server discovery, IP lease offer, IP lease request, and IP lease acknowledgement. These stages are often abbreviated as DORA for discovery, offer, request, and acknowledgement.

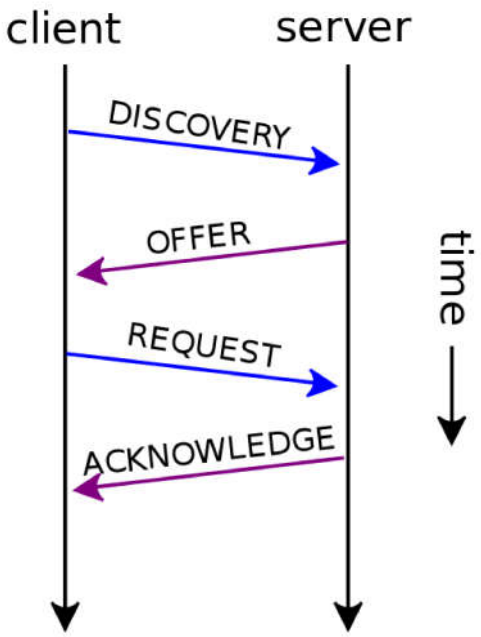
The DHCP operation begins with clients broadcasting a request. If the client and server are on different subnets, a DHCP Helper or DHCP Relay Agent may be used. Clients requesting renewal of an existing lease may communicate directly via UDP unicast, since the client already has an established IP address at that point. Additionally, there is a BROADCAST flag (1 bit in 2 byte flags field, where all other bits are reserved and so are set to 0) the client can use to

indicate in which way (broadcast or unicast) it can receive the DHCPOFFER: 0x8000 for broadcast, 0x0000 for unicast.[10] Usually, the DHCPOFFER is sent through unicast. For those hosts which cannot accept unicast packets before IP addresses are configured, this flag can be used to work around this issue.

DHCP discovery

The DHCP client broadcasts a DHCPDISCOVER message on the network subnet using the destination address 255.255.255.255 (limited broadcast) or the specific subnet broadcast address (directed broadcast). A DHCP client may also request its last known IP address. If the client remains connected to the same network, the server may grant the request. Otherwise, it depends whether the server is set up as authoritative or not. An authoritative server denies the request, causing the client to issue a new request. A non-authoritative server simply ignores the request, leading to an implementation-dependent timeout for the client to expire the request and ask for a new IP address.

For example, if HTYPE is set to 1, to specify that the medium used is Ethernet, HLEN is set to 6 because an Ethernet address (MAC address) is 6 octets long. The CHADDR is set to the MAC address used by the client. Some options are set as well.



An illustration of a typical non-renewing DHCP session; each message may be either a broadcast or a unicast, depending on the DHCP client capabilities.[10]

Example DHCPDISCOVER message

Ethernet: source=sender's MAC; destination=FF:FF:FF:FF:FF:FF			
IP: source=0.0.0.0; destination=255.255.255.255 UDP: source port=68; destination port=67			
Octet 0	**Octet 1**	**Octet 2**	**Octet 3**
OP	**HTYPE**	**HLEN**	**HOPS**
0x01	0x01	0x06	0x00
XID			
0x3903F326			
SECS		**FLAGS**	
0x0000		0x0000	
CIADDR (Client IP address)			
0x00000000			
YIADDR (Your IP address)			
0x00000000			
SIADDR (Server IP address)			
0x00000000			
GIADDR (Gateway IP address)			
0x00000000			
CHADDR (Client hardware address)			
0x00053C04			
0x8D590000			
0x00000000			
0x00000000			
192 octets of 0s, or overflow space for additional options; BOOTP legacy.			
Magic cookie			
0x63825363			
DHCP options			
0x350101 53: 1 (DHCP Discover)			
0x3204c0a80164 50: 192.168.1.100 requested			
0x370401030f06 55 (Parameter Request List): ■ 1 (Request Subnet Mask), ■ 3 (Router), ■ 15 (Domain Name), ■ 6 (Domain Name Server)			
0xff 255 (Endmark)			

DHCP offer

When a DHCP server receives a DHCPDISCOVER message from a client, which is an IP address lease request, the DHCP server reserves an IP address for the client and makes a lease offer by sending a DHCPOFFER message to the client. This message contains the client's client id (traditionally a MAC address), the IP address that the server is offering, the subnet mask, the lease duration, and the IP address of the DHCP server making the offer. The DHCP server may also take notice of the hardware-level MAC address in the underlying transport layer: according to current RFCs the transport layer MAC address may be used if no client ID is provided in the DHCP packet.

The DHCP server determines the configuration based on the client's hardware address as specified in the CHADDR (client hardware address) field. Here the server, 192.168.1.1, specifies the client's IP address in the YIADDR (your IP address) field.

DHCPOFFER message

Ethernet: source=sender's MAC; destination=client mac address			
IP: source=192.168.1.1; destination=255.255.255.255 UDP: source port=67; destination port=68			
Octet 0	**Octet 1**	**Octet 2**	**Octet 3**
OP	**HTYPE**	**HLEN**	**HOPS**
0x02	0x01	0x06	0x00
XID			
0x3903F326			
SECS		**FLAGS**	
0x0000		0x0000	
CIADDR (Client IP address)			
0x00000000			
YIADDR (Your IP address)			
0xC0A80164 (192.168.1.100)			
SIADDR (Server IP address)			
0xC0A80101 (192.168.1.1)			
GIADDR (Gateway IP address)			
0x00000000			
CHADDR (Client hardware address)			
0x00053C04			
0x8D590000			
0x00000000			
0x00000000			
192 octets of 0s; BOOTP legacy.			
Magic cookie			
0x63825363			
DHCP options			
53: 2 (DHCP Offer)			
1 (subnet mask): 255.255.255.0			
3 (Router): 192.168.1.1			
51 (IP address lease time): 86400s (1 day)			
54 (DHCP server): 192.168.1.1			
6 (DNS servers): ▪ 9.7.10.15, ▪ 9.7.10.16,			

- 9.7.10.18

DHCP request

In response to the DHCP offer, the client replies with a DHCPREQUEST message, broadcast to the server,[a] requesting the offered address. A client can receive DHCP offers from multiple servers, but it will accept only one DHCP offer. Based on required *server identification* option in the request and broadcast messaging, servers are informed whose offer the client has accepted.[12]:Section 3.1, Item 3 When other DHCP servers receive this message, they withdraw any offers that they have made to the client and return the offered IP address to the pool of available addresses.

DHCPREQUEST message

Ethernet: source=sender's MAC; destination=FF:FF:FF:FF:FF:FF			
IP: source=0.0.0.0; destination=255.255.255.255;[a] UDP: source port=68; destination port=67			
Octet 0	**Octet 1**	**Octet 2**	**Octet 3**
OP	**HTYPE**	**HLEN**	**HOPS**
0x01	0x01	0x06	0x00
XID			
0x3903F326			
SECS		**FLAGS**	
0x0000		0x0000	
CIADDR (Client IP address)			
0x00000000			
YIADDR (Your IP address)			
0x00000000			
SIADDR (Server IP address)			
0xC0A80101 (192.168.1.1)			
GIADDR (Gateway IP address)			
0x00000000			
CHADDR (Client hardware address)			
0x00053C04			
0x8D590000			
0x00000000			
0x00000000			
192 octets of 0s; BOOTP legacy.			
Magic cookie			
0x63825363			
DHCP options			
53: 3 (DHCP Request)			
50: 192.168.1.100 requested			
54 (DHCP server): 192.168.1.1			

DHCP acknowledgement

When the DHCP server receives the DHCPREQUEST message from the client, the configuration process enters its final phase. The acknowledgement phase involves sending a DHCPACK packet to the client. This packet includes the lease duration and any other configuration information that the client might have requested. At this point, the IP

configuration process is completed.

The protocol expects the DHCP client to configure its network interface with the negotiated parameters.

After the client obtains an IP address, it should probe the newly received address[13] (e.g. with ARP Address Resolution Protocol) to prevent address conflicts caused by overlapping address pools of DHCP servers.

DHCPACK message

Ethernet: source=sender's MAC; destination=client's MAC			
IP: source=192.168.1.1; destination=192.168.1.100 UDP: source port=67; destination port=68			
Octet 0	**Octet 1**	**Octet 2**	**Octet 3**
OP	**HTYPE**	**HLEN**	**HOPS**
0x02	0x01	0x06	0x00
XID			
0x3903F326			
SECS		**FLAGS**	
0x0000		0x0000	
CIADDR (Client IP address)			
0x00000000			
YIADDR (Your IP address)			
0xC0A80164 (192.168.1.100)			
SIADDR (Server IP address)			
0xC0A80101 (192.168.1.1)			
GIADDR (Gateway IP address switched by relay)			
0x00000000			
CHADDR (Client hardware address)			
0x00053C04			
0x8D590000			
0x00000000			
0x00000000			
192 octets of 0s. BOOTP legacy			
Magic cookie			
0x63825363			
DHCP options			
53: 5 (DHCP ACK) or 6 (DHCP NAK)			
1 (subnet mask): 255.255.255.0			
3 (Router): 192.168.1.1			
51 (IP address lease time): 86400s (1 day)			
54 (DHCP server): 192.168.1.1			
6 (DNS servers): ▪ 9.7.10.15,			

- 9.7.10.16,
- 9.7.10.18

DHCP information

A DHCP client may request more information than the server sent with the original DHCPOFFER. The client may also request repeat data for a particular application. For example, browsers use *DHCP Inform* to obtain web proxy settings via WPAD.

DHCP releasing

The client sends a request to the DHCP server to release the DHCP information and the client deactivates its IP address. As client devices usually do not know when users may unplug them from the network, the protocol does not mandate the sending of *DHCP Release*.

Client configuration parameters

A DHCP server can provide optional configuration parameters to the client. RFC 2132 describes the available DHCP options defined by Internet Assigned Numbers Authority (IANA) - DHCP and BOOTP PARAMETERS.[14]

A DHCP client can select, manipulate and overwrite parameters provided by a DHCP server. In Unix-like systems this client-level refinement typically takes place according to the values in the configuration file */etc/dhclient.conf*.

DHCP options

Options are octet strings of varying length. The first octet is the option code, the second octet is the number of following octets and the remaining octets are code dependent. For example, the DHCP message-type option for an offer would appear as 0x35, 0x01, 0x02, where 0x35 is code 53 for "DHCP message type", 0x01 means one octet follows and 0x02 is the value of "offer".

Documented in RFC 2132

The following tables list the available DHCP options, as listed in RFC 2132[15] and IANA registry.[14]

RFC 1497 (BOOTP Vendor Information Extensions) vendor extensions[15]:Section 3

Code	Name	Length	Notes
0	Pad[15]:Section 3.1	0 octets	Can be used to pad other options so that they are aligned to the word boundary; is not followed by length byte
1	Subnet mask[15]:Section 3.3	4 octets	Must be sent before the router option (option 3) if both are included
2	Time offset[15]:Section 3.4	4 octets	
3	Router	Multiples of 4 octets	Available routers, should be listed in order of preference
4	Time server	Multiples of 4 octets	Available time servers to synchronise with, should be listed in order of preference
5	Name server	Multiples of 4 octets	Available IEN 116 name servers, should be listed in order of preference
6	Domain name server	Multiples of 4 octets	Available DNS servers, should be listed in order of preference
7	Log server	Multiples of 4 octets	Available log servers, should be listed in order of preference.
8	Cookie server	Multiples of 4 octets	*Cookie* in this case means "fortune cookie" or "quote of the day", a pithy or humorous anecdote often sent as part of a logon process on large computers; it has nothing to do with cookies sent by websites.
9	LPR Server	Multiples of 4 octets	
10	Impress server	Multiples of 4 octets	
11	Resource location server	Multiples of 4 octets	
12	Host name	Minimum of 1 octet	
13	Boot file size	2 octets	Length of the boot image in 4KiB blocks
14	Merit dump file	Minimum of 1 octet	Path where crash dumps should be stored
15	Domain name	Minimum of 1 octet	
16	Swap server	4 octets	
17	Root path	Minimum of 1 octet	

Code	Name	Length	Notes
18	Extensions path	Minimum of 1 octet	
255	End	0 octets	Used to mark the end of the vendor option field

IP layer parameters per host[15]:Section 4

Code	Name	Length	Notes
19	IP forwarding enable/disable	1 octet	
20	Non-local source routing enable/disable	1 octet	
21	Policy filter	Multiples of 8 octets	
22	Maximum datagram reassembly size	2 octets	
23	Default IP time-to-live	1 octet	
24	Path MTU aging timeout	4 octets	
25	Path MTU plateau table	Multiples of 2 octets	

IP Layer Parameters per Interface[15]:Section 5

Code	Name	Length	Notes
26	Interface MTU	2 octets	
27	All subnets are local	1 octet	
28	Broadcast address	4 octets	
29	Perform mask discovery	1 octet	
30	Mask supplier	1 octet	
31	Perform router discovery	1 octet	
32	Router solicitation address	4 octets	
33	Static route	Multiples of 8 octets	A list of destination/router pairs

Link layer parameters per interface[15]:Section 6

Code	Name	Length	Notes
34	Trailer encapsulation option	1 octet	
35	ARP cache timeout	4 octets	
36	Ethernet encapsulation	1 octet	

TCP parameters[15]:Section 7

Code	Name	Length	Notes
37	TCP default TTL	1 octet	
38	TCP keepalive interval	4 octets	
39	TCP keepalive garbage	1 octet	

Application and service parameters[15]:Section 8

Code	Name	Length	Notes
40	Network information service domain	Minimum of 1 octet	
41	Network information servers	Multiples of 4 octets	
42	Network Time Protocol (NTP) servers	Multiples of 4 octets	
43	Vendor-specific information	Minimum of 1 octets	
44	NetBIOS over TCP/IP name server	Multiples of 4 octets	
45	NetBIOS over TCP/IP datagram Distribution Server	Multiples of 4 octets	
46	NetBIOS over TCP/IP node type	1 octet	
47	NetBIOS over TCP/IP scope	Minimum of 1 octet	
48	X Window System font server	Multiples of 4 octets	
49	X Window System display manager	Multiples of 4 octets	
64	Network Information Service+ domain	Minimum of 1 octet	
65	Network Information Service+ servers	Multiples of 4 octets	
68	Mobile IP home agent	Multiples of 4 octets	
69	Simple Mail Transfer Protocol (SMTP) server	Multiples of 4 octets	
70	Post Office Protocol (POP3) server	Multiples of 4 octets	
71	Network News Transfer Protocol (NNTP) server	Multiples of 4 octets	
72	Default World Wide Web (WWW) server	Multiples of 4 octets	
73	Default Finger protocol server	Multiples of 4 octets	
74	Default Internet Relay Chat (IRC) server	Multiples of 4 octets	
75	StreetTalk server	Multiples of 4 octets	
76	StreetTalk Directory Assistance (STDA) server	Multiples of 4 octets	

DHCP extensions[15]:Section 9

Code	Name	Length	Notes
50	Requested IP address	4 octets	
51	IP address lease time	4 octets	
52	Option overload	1 octet	
53	DHCP message type	1 octet	
54	Server identifier	4 octets	
55	Parameter request list	Minimum of 1 octet	
56	Message	Minimum of 1 octet	
57	Maximum DHCP message size	2 octets	
58	Renewal (T1) time value	4 octets	
59	Rebinding (T2) time value	4 octets	
60	Vendor class identifier	Minimum of 1 octet	
61	Client-identifier	Minimum of 2 octets	
66	TFTP server name	Minimum of 1 octet	
67	Bootfile name	Minimum of 1 octet	

DHCP client vendor identification

An option exists to identify the vendor and functionality of a DHCP client. The information is a variable-length string of characters or octets which has a meaning specified by the vendor of the DHCP client. One method that a DHCP client can utilize to communicate to the server that it is using a certain type of hardware or firmware is to set a value in its DHCP requests called the Vendor Class Identifier (VCI) (Option 60).

This method allows a DHCP server to differentiate between the two kinds of client machines and process the requests from the two types of modems appropriately. Some types of set-top boxes also set the VCI (Option 60) to inform the DHCP server about the hardware type and functionality of the device. The value this option is set to gives the DHCP server a hint about any required extra information that this client needs in a DHCP response.

Documented elsewhere

Documented DHCP options

Code	Name	Length	RFC
82	Relay agent information	Minimum of 2 octets	RFC 3046[16]
85	Novell Directory Service (NDS) servers	Minimum of 4 octets, multiple of 4 octets	RFC 2241[17]:Section 2
86	NDS tree name	Variable	RFC 2241[17]:Section 3
87	NDS context	Variable	RFC 2241[17]:Section 4
100	Time zone, POSIX style	Variable	RFC 4833[18]
101	Time zone, tz database style	Variable	RFC 4833[18]
119	Domain search	Variable	RFC 3397[19]
121	Classless static route	Variable	RFC 3442[20]

Relay agent information sub-options

The relay agent information option (option 82)[16] specifies container for attaching sub-options to DHCP requests transmitted between a DHCP relay and a DHCP server.

Relay agent sub-options

Code	Name	Length	RFC
4	Data-Over-Cable Service Interface Specifications (DOCSIS) device class	4 octets	RFC 3256[21]

DHCP relaying

In small networks, where only one IP subnet is being managed, DHCP clients communicate directly with DHCP servers. However, DHCP servers can also provide IP addresses for multiple subnets. In this case, a DHCP client that has not yet acquired an IP address cannot communicate directly with the DHCP server using IP routing, because it does not have a routable IP address, does not know the link layer address of a router and does not know the IP address of the DHCP server.

In order to allow DHCP clients on subnets not directly served by DHCP servers to communicate with DHCP servers, DHCP relay agents can be installed on these subnets. The DHCP client broadcasts on the local link; the relay agent receives the broadcast and transmits it to one or more DHCP servers using unicast. The relay agent stores its own IP address in field *GIADDR* field of the DHCP packet. The DHCP server uses the GIADDR-value to determine the subnet on which the relay agent received the broadcast, and allocates an IP address on that subnet. When the DHCP server replies to the client, it sends the reply to the GIADDR-address, again using unicast. The relay agent then retransmits the response on the local network.

In this situation, the communication between the relay agent and the DHCP server typically uses both a source and destination UDP port of 67.

Reliability

The DHCP ensures reliability in several ways: periodic renewal, rebinding,[12]:Section 4.4.5 and failover. DHCP clients are allocated leases that last for some period of time. Clients begin to attempt to renew their leases once half the lease interval has expired.[12]:Section 4.4.5 Paragraph 3 They do this by sending a unicast *DHCPREQUEST* message to the DHCP server that granted the original lease. If that server is down or unreachable, it will fail to respond to the *DHCPREQUEST*. However, in that case the client repeats the *DHCPREQUEST* from time to time,[12]:Section 4.4.5 Paragraph 8[b] so if the DHCP server comes back up or becomes reachable again, the DHCP client will succeed in contacting it and renew the lease.

If the DHCP server is unreachable for an extended period of time,[12]:Section 4.4.5 Paragraph 5 the DHCP client will attempt to rebind, by broadcasting its *DHCPREQUEST* rather than unicasting it. Because it is broadcast, the *DHCPREQUEST* message will reach all available DHCP servers. If some other DHCP server is able to renew the lease, it will do so at this time.

In order for rebinding to work, when the client successfully contacts a backup DHCP server, that server must have accurate information about the client's binding. Maintaining accurate binding information between two servers is a complicated problem; if both servers are able to update the same lease database, there must be a mechanism to avoid conflicts between updates on the independent servers. A proposal for implementing fault-tolerant DHCP servers was submitted to the Internet Engineering Task Force, but never formalized.[22][c]

If rebinding fails, the lease will eventually expire. When the lease expires, the client must stop using the IP address granted to it in its lease.[12]:Section 4.4.5 Paragraph 9 At that time it will restart the DHCP process from the beginning by broadcasting a `DHCPDISCOVER` message. Since its lease has expired, it will accept any IP address offered to it. Once it has a new IP address (presumably from a different DHCP server) it will once again be able to use the network. However, since its IP address has changed, any ongoing connections will be broken.

Security

The base DHCP does not include any mechanism for authentication.[23] Because of this, it is vulnerable to a variety of attacks. These attacks fall into three main categories:

- Unauthorized DHCP servers providing false information to clients.[24]
- Unauthorized clients gaining access to resources.[24]
- Resource exhaustion attacks from malicious DHCP clients.[24]

Because the client has no way to validate the identity of a DHCP server, unauthorized DHCP servers (commonly called "rogue DHCP") can be operated on networks, providing incorrect information to DHCP clients.[25] This can serve either as a denial-of-service attack, preventing the client from gaining access to network connectivity,[26] or as a man-in-the-middle attack.[27] Because the DHCP server provides the DHCP client with server IP addresses, such as the IP address of one or more DNS servers,[24] an attacker can convince a DHCP client to do its DNS lookups through its own DNS server, and can therefore provide its own answers to DNS queries from the client.[28][29] This in turn allows the attacker to redirect network traffic through itself, allowing it to eavesdrop on connections between the client and network servers it contacts, or to simply replace those network servers with its own.[28]

Because the DHCP server has no secure mechanism for authenticating the client, clients can gain unauthorized access to IP addresses by presenting credentials, such as client identifiers, that belong to other DHCP clients.[25] This also allows DHCP clients to exhaust the DHCP server's store of IP addresses—by presenting new credentials each time it asks for an address, the client can consume all the available IP addresses on a particular network link, preventing other DHCP clients from getting service.[25]

DHCP does provide some mechanisms for mitigating these problems. The Relay Agent Information Option protocol extension (RFC 3046, usually referred to in the industry by its actual number as *Option 82*[30][31]) allows network operators to attach tags to DHCP messages as these messages arrive on the network operator's trusted network. This

tag is then used as an authorization token to control the client's access to network resources. Because the client has no access to the network upstream of the relay agent, the lack of authentication does not prevent the DHCP server operator from relying on the authorization token.[23]

Another extension, Authentication for DHCP Messages (RFC 3118 (https://tools.ietf.org/html/rfc3118)), provides a mechanism for authenticating DHCP messages. As of 2002, RFC 3118 had not seen widespread adoption because of the problems of managing keys for large numbers of DHCP clients.[32] A 2007 book about DSL technologies remarked that:

> there were numerous security vulnerabilities identified against the security measures proposed by RFC 3118. This fact, combined with the introduction of 802.1x, slowed the deployment and take-rate of authenticated DHCP, and it has never been widely deployed.[33]

A 2010 book notes that:

> [t]here have been very few implementations of DHCP Authentication. The challenges of key management and processing delays due to hash computation have been deemed too heavy a price to pay for the perceived benefits.[34]

Architectural proposals from 2008 involve authenticating DHCP requests using 802.1x or PANA (both of which transport EAP).[35] An IETF proposal was made for including EAP in DHCP itself, the so-called EAPoDHCP;[36] this does not appear to have progressed beyond IETF draft level, the last of which dates to 2010.[37]

IETF standards documents

- RFC 2131, Dynamic Host Configuration Protocol
- RFC 2132, DHCP Options and BOOTP Vendor Extensions
- RFC 3046, DHCP Relay Agent Information Option
- RFC 3397, Dynamic Host Configuration Protocol (DHCP) Domain Search Option
- RFC 3942, Reclassifying Dynamic Host Configuration Protocol Version Four (DHCPv4) Options
- RFC 4242, Information Refresh Time Option for Dynamic Host Configuration Protocol for IPv6
- RFC 4361, Node-specific Client Identifiers for Dynamic Host Configuration Protocol Version Four (DHCPv4)
- RFC 4436, Detecting Network Attachment in IPv4 (DNAv4)
- RFC 3442, Classless Static Route Option for Dynamic Host Configuration Protocol (DHCP) version 4

See also

- Boot Service Discovery Protocol (BSDP) – a DHCP extension used by Apple's NetBoot
- Comparison of DHCP server software
- Peg DHCP (RFC 2322)
- Preboot Execution Environment (PXE)
- Reverse Address Resolution Protocol (RARP)
- Rogue DHCP
- UDP Helper Address – a tool for routing DHCP requests across subnet boundaries
- Zeroconf – Zero Configuration Networking

Notes

a. As an optional client behavior, some broadcasts, such as those carrying DHCP discovery and request messages, may be replaced with unicasts in case the DHCP client already knows the DHCP server's IP address.[11]
b. The RFC calls for the client to wait one half of the remaining time until T2 before it retransmits the *DHCPREQUEST* packet
c. The proposal provided a mechanism whereby two servers could remain loosely in sync with each other in such a way that even in the event of a total failure of one server, the other server could recover the lease database and continue operating. Due to the length and complexity of the specification, it was never published as a standard; however, the techniques described in the specification are in wide use, with one open-source implementation in the ISC DHCP server, as well as several commercial implementations.

References

1. TechTarget Network: DHCP (Dynamic Host Configuration Protocol) (http://searchnetworking.techtarget.com/definition/DHCP)
2. Peterson, Larry L.; Davie, Bruce S. (2011). *Computer Networks: A Systems Approach* (https://books.google.com/books?id=BvaFreun1W8C&pg=PA372&lpg=PA372) (5th ed.). Elsevier. ISBN 0123850606. Retrieved March 21, 2019.
3. Bill Croft; John Gilmore (September 1985). "RFC 951 - Bootstrap Protocol" (http://tools.ietf.org/html/rfc951#section-6). *Network Working Group*.
4. Network+ Certification 2006 Published By Microsoft Press.
5. used for the Web Proxy Autodiscovery Protocol Web Proxy Autodiscovery Protocol (WPAD)
6. RFC 4361, RFC 5494, RFC 6221, RFC 6422, RFC 6644, RFC 7083, RFC 7227, RFC 7283
7. Droms, Ralph. "Dynamic Host Configuration Protocol" (https://tools.ietf.org/html/rfc2131). *tools.ietf.org*. Retrieved 4 July 2017.
8. "ISC's open source DHCP software system - Internet Systems Consortium" (http://www.isc.org/products/DHCP/). *www.isc.org*. Retrieved 4 July 2017.
9. Droms, Ralph; Lemon, Ted (2003). *The DHCP Handbook*. SAMS Publishing. p. 436. ISBN 978-0-672-32327-0.
10. Droms, Ralph. "Dynamic Host Configuration Protocol" (https://tools.ietf.org/html/rfc2131#section-4.1). *tools.ietf.org*. Retrieved 4 July 2017.
11. Droms, Ralph. "Dynamic Host Configuration Protocol" (https://tools.ietf.org/html/rfc2131#section-4.4.4). *tools.ietf.org*. Retrieved 4 July 2017.
12. Droms, Ralph (March 1997). *DHCP Options and BOOTP Vendor Extensions* (https://tools.ietf.org/html/rfc2131). IETF. doi:10.17487/RFC2131 (https://doi.org/10.17487%2FRFC2131). RFC 2131. Retrieved September 9, 2014.
13. "RFC2131 Dynamic Host Configuration Protocol: Dynamic allocation of network addresses" (http://tools.ietf.org/html/rfc2131#section-2.2). *tools.ietf.org*.
14. "Dynamic Host Configuration Protocol (DHCP) and Bootstrap Protocol (BOOTP) Parameters" (https://www.iana.org/assignments/bootp-dhcp-parameters/bootp-dhcp-parameters.xhtml). iana.org. Retrieved 2018-10-16.
15. Alexander, Steve; Droms, Ralph (March 1997). *DHCP options and BOOTP vendor extensions* (https://tools.ietf.org/html/rfc2132). IETF. doi:10.17487/RFC2132 (https://doi.org/10.17487%2FRFC2132). RFC 2132. Retrieved June 10, 2012.
16. Patrick, Michael (January 2001). "DHCP Relay Agent Information Option" (https://tools.ietf.org/html/rfc3046). *IETF Documents*. IETF. doi:10.17487/RFC3046 (https://doi.org/10.17487%2FRFC3046). Retrieved 22 July 2017.
17. Provan, Don (November 1997). "RFC 2241 – DHCP Options for Novell Directory Services" (https://tools.ietf.org/html/rfc2241). *IETF Documents*. IETF. doi:10.17487/RFC3256 (https://doi.org/10.17487%2FRFC3256). Retrieved 23 July 2017.
18. Lear, E.; Eggert, P. (April 2007). "Timezone Options for DHCP" (https://tools.ietf.org/html/rfc4833). *IETF Documents*. IETF. Retrieved 28 June 2018.

19. Bernard, Aboba; Stuart, Cheshire (November 2002). "RFC 3397 – Dynamic Host Configuration Protocol (DHCP) Domain Search Option" (https://tools.ietf.org/html/rfc3397). *IETF Documents*. IETF. doi:10.17487/RFC3397 (https://doi.org/10.17487%2FRFC3397). Retrieved 22 July 2017.
20. RFC 3442 (https://tools.ietf.org/html/rfc3442)
21. Doug, Jones; Rich, Woundy (April 2002). "RFC 3256 – The DOCSIS (Data-Over-Cable Service Interface Specifications) Device Class DHCP (Dynamic Host Configuration Protocol) Relay Agent Information Sub-option" (https://tools.ietf.org/html/rfc3256). *IETF Documents*. IETF. doi:10.17487/RFC3256 (https://doi.org/10.17487%2FRFC3256). Retrieved 23 July 2017.
22. Droms, Ralph; Kinnear, Kim; Stapp, Mark; Volz, Bernie; Gonczi, Steve; Rabil, Greg; Dooley, Michael; Kapur, Arun (March 2003). *DHCP Failover Protocol* (https://tools.ietf.org/html/draft-ietf-dhc-failover-12). IETF. I-D draft-ietf-dhc-failover-12. Retrieved May 9, 2010.
23. Patrick, Michael (January 2001). "RFC 3046 - DHCP Relay Agent Information Option" (http://tools.ietf.org/html/rfc3046#section-7). *Network Working Group*.
24. Droms, Ralph (March 1997). "RFC 2131 - Dynamic Host Configuration Protocol" (http://tools.ietf.org/html/rfc2131#section-7). *Network Working Group*.
25. Stapko, Timothy (2011). *Practical Embedded Security: Building Secure Resource-Constrained Systems* (https://books.google.com/books?id=Mly55VntuYMC&pg=PA39). Newnes. p. 39. ISBN 978-0-08-055131-9.
26. Rountree, Derrick (2013). *Windows 2012 Server Network Security: Securing Your Windows Network Systems and Infrastructure* (https://books.google.com/books?id=NFzou_d4MGUC&pg=SA2-PA13). Newnes. p. 22. ISBN 978-1-59749-965-1.
27. Rooney, Timothy (2010). *Introduction to IP Address Management* (https://books.google.com/books?id=QgRDxkuI1MkC&pg=PA180). John Wiley & Sons. p. 180. ISBN 978-1-118-07380-3.
28. Golovanov (Kaspersky Labs), Sergey (June 2011). "TDSS loader now got "legs" " (http://www.securelist.com/en/blog/208188095/TDSS_loader_now_got_legs).
29. Sunny, Akash K (October 2015). "dhcp protocol and its vulnerabilities" (http://greyhatsspeak.blogspot.com/2015/11/dhcp-protocol-and-its-vulnerabilities.html).
30. Hens, Francisco J.; Caballero, José M. (2008). *Triple Play: Building the converged network for IP, VoIP and IPTV* (https://books.google.com/books?id=aS1ZngveBIkC&pg=PA239). John Wiley & Sons. p. 239. ISBN 978-0-470-75439-9.
31. Ramirez, David H. (2008). *IPTV Security: Protecting High-Value Digital Contents* (https://books.google.com/books?id=70tr_hSDULwC&pg=PA55). John Wiley & Sons. p. 55. ISBN 978-0-470-72719-5.
32. Lemon, Ted (April 2002). "Implementation of RFC 3118" (http://www.ietf.org/mail-archive/web/dhcwg/current/msg00876.html).
33. Golden, Philip; Dedieu, Hervé; Jacobsen, Krista S. (2007). *Implementation and Applications of DSL Technology* (https://books.google.com/books?id=Jjkd74jY47oC&pg=PA484). Taylor & Francis. p. 484. ISBN 978-1-4200-1307-8.
34. Rooney, Timothy (2010). *Introduction to IP Address Management* (https://books.google.com/books?id=QgRDxkuI1MkC&pg=PA181). John Wiley & Sons. pp. 181–182. ISBN 978-1-118-07380-3.
35. Copeland, Rebecca (2008). *Converging NGN Wireline and Mobile 3G Networks with IMS* (https://books.google.com/books?id=ruWv8RGkBGgC&pg=PA142). Taylor & Francis. pp. 142–143. ISBN 978-1-4200-1378-8.
36. Prasad, Ramjee; Mihovska, Albena (2009). *New Horizons in Mobile and Wireless Communications: Networks, services, and applications* (https://books.google.com/books?id=w9bEwBwd33MC&pg=PA339). **2**. Artech House. p. 339. ISBN 978-1-60783-970-5.
37. "Archived copy" (https://web.archive.org/web/20150403091552/http://tools.ietf.org/search/draft-pruss-dhcp-auth-dsl-07). Archived from the original (http://tools.ietf.org/search/draft-pruss-dhcp-auth-dsl-07) on 2015-04-03. Retrieved 2013-12-12.
