



Dynamic Host Configuration Protocol (DHCP)




History

- Diskless workstations
 - needed to know configuration parameters like IP address, netmask, gateway address etc. on boot
 - Small boot programs on ROM needed to load OS over network (*bootstrap*), need to know boot file server
 - Knows only its h/w address on boot
- **RARP** (Reverse ARP) can provide some parameters, but complex
- **BOOTP** (Bootstrap Protocol) designed to provide these, and some more
- Main problem with BOOTP – no dynamic IP allocation



DHCP

- Extension of BOOTP – allows dynamic IP allocation
- Similar message format as BOOTP – DHCP server compatible with BOOTP clients
- Can give a wide range of configuration parameters to hosts
 - IP address, netmask, router, DNS server, boot file server, boot file name, time server, domain name, host name, SMTP server, POP3 server, default IP TTL etc. (see RFC 2132)

- 
- Server on UDP port 67, client on UDP port 68 (not ephemeral)
 - RFC 2131

Motivation

- Move a m/c between networks
- Add a new m/c
- Reclaim unused IP address space
- Configuration of the m/c should be simple and automatic

IP Address Allocation


- Automatic Allocation
 - Permanent address given to host by DHCP, never expires
- Dynamic Allocation
 - IP address given to host for specific time (*lease*)
 - Client host can relinquish before that time or renew after that time
 - Most common allocation method
- Manual
 - Allocated by administrator (specified in a file)
 - DHCP only conveys that information to host


Basic Operation


- Client sends **DHCPDISCOVER** to DHCP server
- Server replies with **DHCPOFFER** with offer of parameters
- If client accepts, client sends **DHCPREQUEST** to server
- Server commits the offer to storage, sends **DHCPACK**. If any problem, don't commit and send **DHCPNACK**, client starts again
- Client checks again. If accept, do nothing. If problems, send **DHCPDECLINE** to server
- Client can release parameters before lease time expiry by **DHCPRELEASE**


Basic Operation – Address Allocation

- Client broadcasts **DHCPDISCOVER** message on local subnet
 - Use 255.255.255.255 as destination IP
 - Source IP set to 0
 - Client's h/w id sent in message for identification
 - Client can request a specific IP and specific lease time
 - Client can specify a preferred DHCP server
 - Clients can specify what parameters are desired (may not want all parameters that the server can give)

- 
- All DHCP servers, on receiving the DHCPDISCOVER
 - Allocates IP for client
 - Allocate IP requested by client if possible
 - Dynamically allocate an IP from pool of available IPs
 - Lease time set to whatever client has requested (subject to local policy) or default
 - Selects parameters requested by client, or default set if no specific request
 - Send a DHCP OFFER message to client. This is an offer, not a final commitment.

- 
- Message either unicast to h/w address of client or broadcast
 - Client, on receiving DHCP OFFER
 - Can receive offer from multiple servers
 - Select one server, accept its offer, and send a DHCPREQUEST to it with the configuration parameters
 - Request is broadcast, but a special server identifier field filled with the server chosen, so that other servers know they are being declined
 - Timeout to start rediscovery if no offer within a certain time
 - DHCPREQUEST also sent on boot to confirm earlier allocation, or to renew earlier lease

- 
- Server, on receiving DHCPREQUEST
 - If offer is still valid, commit offer to persistent storage and send DHCPACK to client with the configuration parameters (same as in offer)
 - If unable to satisfy request now, send DHCPNAK
 - Makes earlier offered address available for others if timeout for request

- 
- Client on receiving DHCPACK/NAK
 - If NAK, start entire process again
 - If ACK, make final check. If no problem, do nothing. If problem, send DHCPDECLINE
 - Send request again if timeout for ACK/NAK, give up after few tries
 - Client can release an IP before its lease expiry by DHCPRELEASE message
 - If client has already got IP externally (ex. static IP), but wants other config parameters, should use DHCPINFORM message

DHCP Message Format

0	8	16	24	31
OP	HTYPE	HLEN	HOPS	
TRANSACTION ID				
SECONDS		FLAGS		
CLIENT IP ADDRESS				
YOUR IP ADDRESS				
SERVER IP ADDRESS				
ROUTER IP ADDRESS				
CLIENT HARDWARE ADDRESS (16 OCTETS)				
⋮				
SERVER HOST NAME (64 OCTETS)				
⋮				
BOOT FILE NAME (128 OCTETS)				
⋮				
OPTIONS (VARIABLE)				
⋮				

Figure 23.5 The format of a DHCP message, which is an extension of a BOOTP message. The options field is variable length; a client must be prepared to accept at least 312 octets of options.

- **operation:** 1 = request, 2 = reply
- **htype:** hardware type (ethernet etc., integer code for defined types)
- **hlen:** length of h/w address
- **xid:** transaction id used to match request and reply
- **secs:** time since client begun address acquisition/renewal
- **flags:** top bit indicates if client wants Broadcast from server or not
- **ciaddr:** Client IP address. Filled in by client in some cases when client knows its IP and can respond to ping
- **yiaddr:** IP address of client, Filled in by server after allocation/renewal
- **siaddr:** IP of server to handle the next stage
- **giaddr:** IP of relay agent, if any. 0 indicates no relay
- **chaddr:** client h/w address
- **sname:** optional DHCP server name to contact
- **file:** boot file name
- **options:** other configuration parameters

Some Examples of Field Settings

- DHCPDISCOVER
 - op = 1 (request)
 - xid = selected by client
 - secs = 0 or seconds since DHCP process started (if timeout etc.)
 - flags = set BROADCAST bit if client requires broadcast replies
 - ciaddr = 0
 - yiaddr = 0
 - siaddr = 0
 - giaddr = 0
 - chaddr = client's hardware address
 - sname = usually unused, can be used as option

- DHCPOFFER
 - op = 2 (reply)
 - secs = 0
 - flags = flags from client DHCPDISCOVER
 - xid = xid from client DHCPDISCOVER
 - ciaddr = 0
 - yiaddr = offered IP address
 - siaddr = IP address of next server to service request
 - giaddr = giaddr from client DHCPDISCOVER
 - chaddr = chaddr from client DHCPDISCOVER
 - Sname = server's host name or option



- DHCPREQUEST

- op = 1 (request)
- xid = xid from DHCPOFFER
- secs = 0 or seconds since DHCP process started (if timeout etc.)
- flags = set BROADCAST bit if client requires broadcast replies
- ciaddr = 0 for initial allocation, client's IP if renewing/rebinding
- yiaddr = 0
- siaddr = 0
- giaddr = 0
- chaddr = client's hardware address
- sname = usually unused, can be used as option



- Important point

- If ciaddr field is filled in, client must be ready to accept packets to that IP (for ex., reply to ping) and reply to ARP requests for that IP



Options

- specifies additional configuration parameters
- First 4 octets are special – contains decimal 99, 130, 83, 99 (*magic cookie*)
- Variable length list of options after that, always ends with a special option “end”
- Format of each option – TLV encoded
 - 1 octet type code (integer code for each option)
 - 1 octet length
 - Multioctet value of specified length



Example Options

- Message type – type of message (discover, request, offer etc.)
- Client identifier – an explicit identifier specified by the client. Client must use the same identifier in all subsequent messages
- Requested IP address – set by client to request a specific IP
- Maximum DHCP message size – set by client to indicate maximum size of message acceptable to it
- Parameter request list – list of config parameters requested by client
- Server identifier – server selected by client among multiple offers, sent in DHCPREQUEST. Also sent in all server-to-client messages with server's IP
- See RFC 2132 for a list of options
- New options can be added through central authority (see RFC 2939)



Persistent Storage in DHCP

- Once an allocation is committed, it needs to be stored to make it persistent
 - Needs unique id for a client
 - Needs unique id for a lease
- Client id
 - If client supplied an explicit id in client identifier option, use it
 - Else, (IP subnet number, h/w address specified in *ciaddr* field)
 - This implies host with multiple interfaces must do a separate DHCP discover etc. for each interface



- Lease id
 - Needed to identify a lease for renewal etc.
 - Id for lease = (client's id, assigned network address)
- DHCP tries to assign the same address to a client everytime it requests
 - Looks up last lease of client if there in storage, if address is unallocated, assign that
 - Overridden by "requested IP address" option if set by client



Lease Renew/Rebind

- Start renewal process before expiry (by default, after 50% lease time is over)
 - Contact original DHCP server
 - Start from DHCPREQUEST step, fill in *ciaddr* field
 - Simple UDP communication, since client is fully configured
- If no response to renewal after certain time (by default, after 87.5% of lease time)
 - Contact any DHCP server for renewal (broadcast)
 - Still starts from DHCPREQUEST and uses plain UDP communication
- If lease expires, must start from DHCPDISCOVER again



BOOTP/DHCP Relay

- At least one DHCP server per subnet may be too costly if no. of subnets is large
- Relay agents: special processing for packets to UDP port 67
- *giaddr* field: 0 indicates client-server on same subnet, no relay. If not, field specifies IP address of relay agent
 - Set to 0 by client. First relay agent that picks up the DHCP packet from client puts its IP in that field, not changed by subsequent relays
- Any message from server to client sent to relay agent's DHCP server port if *giaddr* is non-zero
- See BOOTP Relay part of RFC 1542