

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 (BootstrapProtocol) designed to provide these, and some more
- Main problem with BOOTP – no dynamic IP allocation
 - IP allocated to a device is fixed at BOOTP server (based on its MAC)

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)
- Runs on UDP
 - Server on UDP port 67
 - Client on UDP port 68
 - But UDP is encapsulated in IP, and IP header will need source IP address
 - How can a device use UDP before it actually gets an IP?

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
- Primarily used for network configuration
 - IP, subnet, gateway, DNS
 - Other parameters can be given also as mentioned

DHCP based 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
 - All 1's is limited broadcast IP for broadcast within subnet
 - Source IP set to 0
 - Client's MAC 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 (DHCP pool configured in the server)
 - 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 DHCPOFFER message to client. This is an offer, not a final commitment.
 - Message either unicast to h/w address of client or broadcast. Client should specify broadcast (through a flag bit) in any message to server if it cannot handle unicast before TCP/IP configuration

- Client, on receiving DHCPOFFER
 - 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 declines
 - 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 (for ex., is IP in use?). 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 configuration parameters, should use DHCPINFORM message

DHCP Message Format

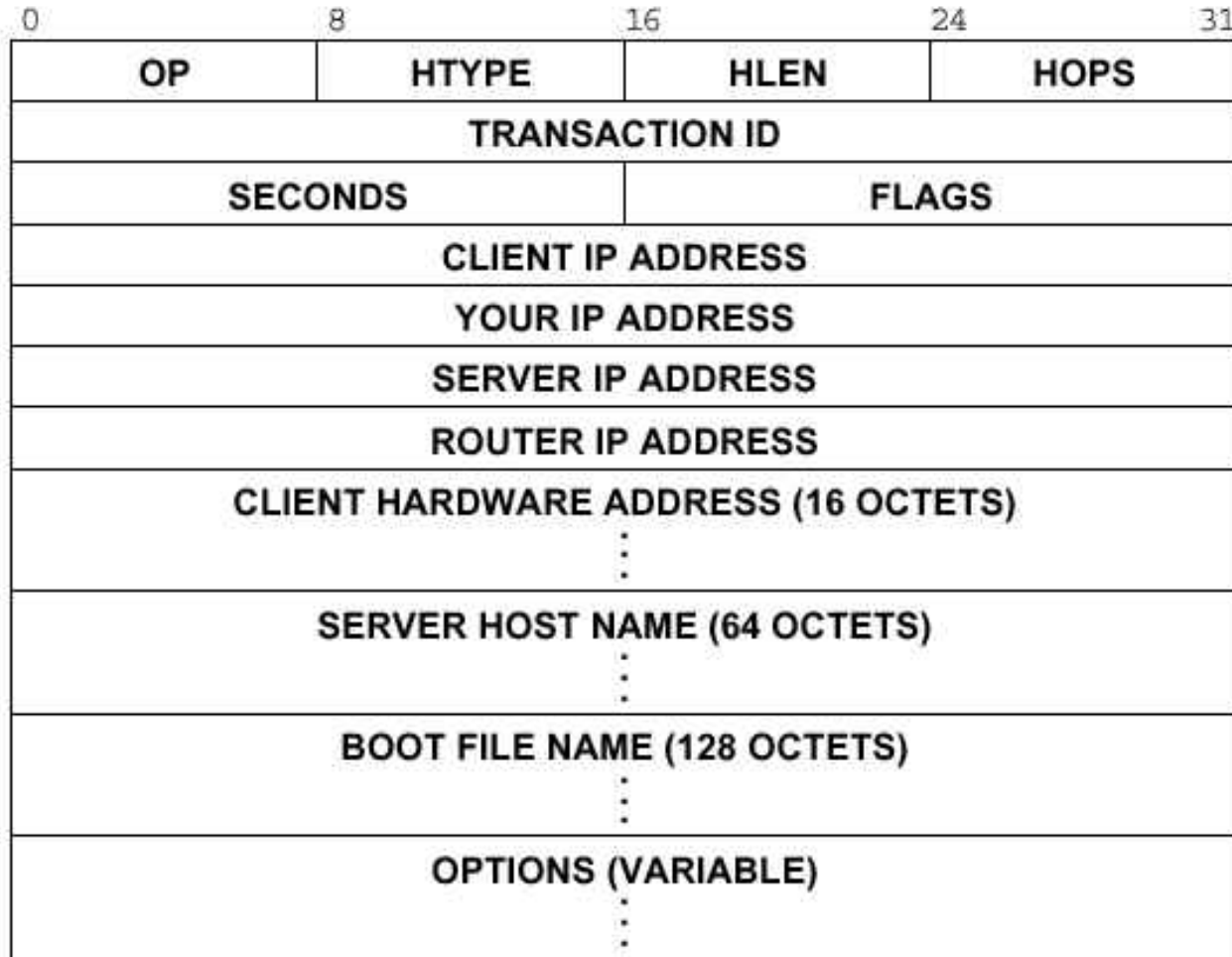


Figure 23.5 The format of a DHCP message, which is an extension of a BOOTP message. The options field is variable length; a client must be prepared to accept at least 312 octets of options.

- *operation*: 1 = request, 2 = reply
- *htype*: hardware type (ethernet etc., integer code for defined types)
- *hlen*: length of h/w address
- *hops*: no. of routers the message passes through
- *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 (for ex., when renewing a lease)
- *yiaddr*: IP address of client. Filled in by server after allocation/renewal
- *siaddr*: IP of server to handle the next stage. Filled in by server in reply, for ex., in a DHCPOFFER
- *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*: message type and other parameters

- DHCP options
 - Options field is not really optional (so cannot have a DHCP message with no options)
 - All DHCP messages must contain the “Message Type” option, which identifies what message it is
 - Depending on the message type, additional options are allowed (can be there), required (must be there), not allowed (cannot be there). Simple examples:
 - DHCPDISCOVER can contain a “Requested IP Address” and “IP Address Lease Time” options to request a specific IP to be allocated and a specific lease time
 - DHCPREQUEST will contain the DHCP server chosen in the “Server Identifier” option
 - Long list, with different constraints. See RFC 2132 for details

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 chaddr 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, before or after lease expiry
 - 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 Renewal

- 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 with current IP already allocated
 - 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
- Suppose that client machine and DHCP server are in different subnets
- DHCPDISCOVER message will not be forwarded by the router
 - Limited broadcast (all 1 destination IP) is only within the subnet
 - So client's request will not reach the server
- Solution: make the router act as a **DHCP relay agent**
 - Routers can be configured to give special processing to DHCP packets even if the destination IP is a broadcast IP

- Configure/Enable the router to act as relay agent
 - Also configures the DHCP server addresses in the relay agent
 - Can configure multiple DHCP servers
- Client sets a field (*giaddr*) to 0 when it sends the request
 - 0 indicates client-server on same subnet
- Relay agent checks for packets with destination UDP port 67
- If detected,
 - First relay agent detecting this (*giaddr* = 0) puts its own IP (IP of interface the packet was received in) in this field
 - Not changed by subsequent relays
 - Sends the UDP packet to each of the DHCP servers
 - Source IP = relay agent IP, destination IP = DHCP server IP, source port = 68, destination port = 67
 - UDP packet forwarded normally

- So in essence the first relay agent acts as a proxy client to the DHCP servers
- DHCP server, on receiving a packet, checks the *giaddr* field
 - If 0, client on the same subnet, send as earlier
 - If non-zero, send response (DHCPOFFER) to relay agent
 - Source IP = DHCP server IP, destination IP = relay agent IP (from *giaddr* field), source port = 67, destination port = 68
 - UDP packet forwarded normally
- Relay agent forwards to original client
- Same for all other subsequent messages