DHCPv6

Shane Kerr (ISC), October 2006

Abstract

This presentation discusses DHCPv6 and how it differs from DHCP in IPv4. An explanation of why you might use DHCPv6 instead of IPv6 stateless auto-configuration is given. Finally, there are some observations from the recent ISC DHCPv6 implementation work.

Contents

What is DHCP?

DHCP is a protocol that allows computers to get configuration information about the network from the network. Addresses are "leased" from servers to clients for a period of time.

Here's a quick, simplified summary of how it works:

- When a computer starts, it sends a request to the network.
- Any DHCP servers that receive this request decide what address and other information to assign to the client. This is typically based on things like: which network the request arrived on, or the MAC address of the Ethernet card that sent the request.
- Each server sends a packet which offers to assign the address to the client.
- The client decides which offer to accept, and sends a message to the server confirming the choice.
- The server acknowledges that it has recorded this address.

At this point the client has an address, and usually other information like netmask, default router, DNS server and default domain, and so on. This information is stored in "options". The ability to add options for additional information or features is one of the great strengths of DHCP.

DHCP supports relays. This is useful both to avoid having to carry broadcast requests throughout your network, and also to consolidate to a small number of servers rather than putting one on each segment.

DHCP also has a failover mechanism, which allows two servers to share a pool of

addresses. This is primarily for redundancy, but also allows load balancing.

How is DHCPv6 different from DHCP in IPv4?

The protocol summary above is basically the same for both DHCP in IPv4 and for DHCPv6. However, the details of DHCPv6 are very, very different from DHCP in IPv4.

- No baggage.
  - DHCP is based on an earlier protocol called BOOTP. This packet layout is wasteful in a lot of cases.
  - A lot of the options turn out to be not useful, or not as useful as they can be, but it is hard to change a protocol with such a large installed base.
  - There are a lot of "tweaks" that implementations need in order to be compatible with the buggy clients.

  DHCPv6 leaves all this behind.
- IPv6 is better.
  Two features of IPv6 greatly improve DHCPv6:
  - IPv6 hosts have "link-local addresses". Every network interface has a unique address, that can be used to send and receive on the link only. IPv6 hosts can use this to send requests for "real" addresses. IPv4 hosts have to use system-specific hacks to work before they have an address.
  - All IPv6 systems support multicasting. All DHCPv6 servers register that they want to recieve DHCPv6 multicast packets. This means the network knows where to send them. In IPv4, clients broadcast their requests, and networks do not know how far to send them.
- One exchange configures all interfaces.
  A single DHCPv6 request may includes all interfaces on a client. This allows the server to offer addresses to all interfaces in a single exchange. Each interface may also have different options.
- Defines address allocation types.
  DHCPv6 allows normal address allocation, as well as temporary address allocation. In a sense, all addresses are "temporary", but the in this case it means the IPv6 privacy addresses.

DHCPv6 does not have as many options defined as DHCP for IPv4, but there are quite a few. You can find these by searching the IETF RFCs, and they include:

IPv6 address, IPv6 prefix
Rapid commit
Vendor-specific options extension
SIP servers
DNS servers & search options
NIS configuration
SNTP servers

BCMS servers

Why would I use DHCPv6, rather than autoconfiguration?

One of the "gee, cool" features of IPv6 is that hosts can get an address automatically. The way this works is that the router on the network periodically broadcasts the address that the network uses. The hosts then add unique information on to part of that, and the combination is a new IPv6 address.

There are two problems with this scenerio:

1. IPv6 address autoconfiguration provides a very limited amount of information about the network.
   For instance, until very recently, there was no way to find out which addresses provided DNS (as of August 2006 there is an experimental router advertisement that provides DNS information)[1].
   DHCPv6 is easily extended to include new information as the need arises.
2. In a managed network, you may prefer to have control over which addresses are allocated. Even if you don't care which addresses are allocated, knowing when a given host used a given address is useful for audit purposes.
   Also, a DHCPv6 server can perform certain tasks that you don't want any machine on the network doing. An example of this is using DDNS to update the reverse DNS information for the IPv6 address.

In the end, the question of "stateless autoconfiguration vs. DHCPv6" is probably a bad question. It is possible to use autoconfiguration to set host addresses, and then use DHCPv6 to provide further information to the hosts.

Implementation observations

ISC has been busy implementing DHCPv6 for a couple of months now. So, I'd like to share a few observations from the implementor side.

- DHCPv6 is much, much cleaner than DHCP for IPv4.
  - All of the differences outlined in the comparison section make for a cleaner protocol.
  - Not having to hack below the socket layer in the OS is a beautiful thing.

  The RFCs and other documentation are even easy to read!
- DHCPv6 is a different world from DHCP for IPv4.
  - A DHCPv6 packet may have options encapsulated at many different levels. In the outermost packet, in each interface, within each address, and so on.
  - Part of a request can fail while other parts succeed.
  - The same option can appear multiple times, with different values, within a single encapsulation level.

1 http://www.ietf.org/internet-drafts/draft-jeong-dnsop-IPv6-dns-discovery-09.txt

Being cleaner doesn't necessarily mean simpler. :)

Existing implementations

There are a number of DHCPv6 implementations that you can use. Note that this list is mostly just the results of a bit of research with Google.

- ISC DHCP
  This is actually a lie, because we don't have a release yet. We are currently working on both a DHCPv6 client and server, but these are not expected to be released until early 2007. We aim to be a reference implemention for DHCPv6, the same as for DHCP and DNS, targetting a wide range of Unix platforms. BSD license.
- DHCPv6: Dibbler
  Appears to be an active project, ported to a number of systems. Supports Linux and Windows. GPL license.
- WIDE-DHCPv6
  DHCPv6 from the KAME project. Appears to be active. BSD license.
- DHCPv6 @ SourceForge
  Appears to be an inactive project, last release 2004-03-15. BSD license.
- Cisco IOS
  Cisco seems to have a DHCPv6 server.

There are surely other clients and servers. If you want to run DHCPv6, the tools are there.

Conclusion

DHCPv6 has a place in IPv6 networks. It is significantly improved over DHCP in IPv4, and is useful either instead of or in addition to stateless autoconfiguration. The software is there today, and getting better, so seriously consider using it.