            Dynamic Host Configuration Protocol (DHCP):
                    IPv4 and IPv6 Dual-Stack Issues

Status of This Memo

   This memo provides information for the Internet community. It does
   not specify an Internet standard of any kind. Distribution of this
   memo is unlimited.

Copyright Notice

Abstract

   A node may have support for communications using IPv4 and/or IPv6
   protocols. Such a node may wish to obtain IPv4 and/or IPv6
   configuration settings via the Dynamic Host Configuration Protocol
   (DHCP). The original version of DHCP (RFC 2131) designed for IPv4
   has now been complemented by a new DHCPv6 (RFC 3315) for IPv6. This
   document describes issues identified with dual IP version DHCP
   interactions, the most important aspect of which is how to handle
   potential problems in clients processing configuration information
   received from both DHCPv4 and DHCPv6 servers. The document makes a
   recommendation on the general strategy on how best to handle such
   issues and identifies future work to be undertaken.

Table of Contents

1. Introduction

   The original specification of the Dynamic Host Configuration Protocol
   (DHCP) was made with only IPv4 in mind. That specification has been
   subsequently revised, up to the latest version of DHCP [1]. With the
   arrival of IPv6, a new DHCP specification for IPv6 has been designed
   and published as DHCPv6 [4].

   These protocols allow nodes to communicate via IPv4 or IPv6
   (respectively) to retrieve configuration settings for operation in a
   managed environment. While an IPv6 node may acquire address-related
   configuration settings via IPv6 stateless address autoconfiguration
   [2], such a node may wish to use stateless DHCPv6 [5] for other
   administratively configured options, such as DNS or NTP.

   In early IPv6 deployments, a dual-stack mode of operation is
   typically used. There will thus be nodes that require both IPv4 and
   IPv6 configuration settings. This document discusses issues with
   obtaining such settings in a dual-stack environment.

   There is a general multihoming issue to be solved for DHCP. A host
   might simultaneously be connected to multiple networks managed by
   multiple parties. Also, IPv4 and IPv6 might be managed by separate
   parties. While these issues are touched on in this document, here we
   focus on the specific issues for operating DHCP in a mixed (typically
   dual-stack) IPv4 and IPv6 environment within a single administrative
   domain.

   In this document, we refer to a "DHCP server" as a server
   implementing the original DHCP [1], and a "DHCPv6 server" as a server
   implementing DHCPv6 [4] or its stateless subset [5].

2. Configuration Scenarios

   For a node in an IPv4-only or IPv6-only environment, the choice of
   DHCP server is a straightforward one; a DHCP server for IPv4, or a
   DHCPv6 server for IPv6.

   In a dual-stack environment a node in a managed environment will need
   to obtain both IPv4 and IPv6 configuration settings, such as the
   following:

   o IPv4 address

   o IPv6 address

   o NTP server

   o DNS server

   o NIS server

   o DNS search path

   While the format of address settings will be IP specific, the node
   may equally well acquire IPv4 or IPv6 addresses for some settings,
   such as for DNS or NTP, if those services are available via IPv4 or
   IPv6 transport. Currently, a DHCP server returns IPv4 data, while a
   DHCPv6 server returns IPv6 data.

   It is worth noting that in an IPv4 environment, with a DHCP server,
   the choice of whether to use DHCP is made by the node. In an IPv6
   environment, the use of the managed and other bits in the Router
   Advertisement can offer a hint to the node whether or not to use full
   DHCPv6 or its stateless variant. It is perhaps not clear whether a
   dual-stack node should do DHCP for IPv4 if Managed and OtherConfig
   flags in the Router Advertisement are both off; it seems most
   appropriate that the decision to use DHCP for IPv4 or not should be
   as if the host were IPv4-only.

3. Dual-Stack Issues

   In this section, we list issues that have been raised to date,
   related to dual-stack DHCP operation.

   It has been noted from comments that the first four, and possibly
   five, subsections here may also be viewed as multihoming issues.

3.1. Handling Multiple Responses

   The general question is how to handle configuration information that
   may be gathered from multiple sources. Where those sources are DHCP
   and DHCPv6 servers (which may be two physical nodes or two servers
   running on the same node) the client node needs to know whether to
   use the most recent data, or whether to perform some merger or union
   of the responses by certain rules. A method for merging lists of
   addresses, for options that carry such information, may also be
   required. A node may choose to ask a DHCPv6 server and only use a
   DHCP server if no response is received.

   Merging is possible, but is likely to be complex. There could be
   some priority, so that if both DHCP and DHCPv6 servers offer a value,
   only one is used. Or the node could choose to store and use both, in
   some order of its choosing. Merging issues are further discussed
   later in this document.

A node may also obtain information from other sources, such as a
manual configuration file (for example, /etc/resolv.conf for DNS data
on many UNIX systems). A node configured manually to use an IPv6 DNS
server may lose that configuration if it is in a dual-stack
environment and uses DHCP to obtain IPv4 settings; the new IPv4
settings from the DHCP response may then overwrite the manual IPv6
DNS setting.

3.2. Different Administrative Management

   In some deployments, the IPv4 and IPv6 services may not be
   administered by the same organisation or people, such as in a
   community wireless environment. This poses problems for consistency
   of data offered by either DHCP version.

   There may also be different connectivity for the protocols, and the
   client may gain advantage from knowing which 'administrative domain'
   is supplying which information. A client may need to use different
   received information depending on which connectivity is being used.
   In the example of the community wireless environment, the question of
   which connectivity is 'better' is a separate issue.

3.3. Multiple Interfaces

   A node may have multiple interfaces and run IPv4 and IPv6 on
   different interfaces. A question then is whether the settings are
   per interface or per node.

   Per-interface settings can be complex because a client node needs to
   know which interface system settings, like NTP server, came from.
   And it may not be apparent which setting should be used if, for
   example, an NTP server option is received on multiple interfaces,
   potentially over different protocols.

3.4. DNS Load Balancing

   In some cases it is preferable to list DNS server information in an
   ordered way per node for load balancing, giving different responses
   to different clients. Responses from different DHCP and DHCPv6
   servers may make such configuration problematic, if the knowledge of
   the load balancing is not available to both servers.

3.5. DNS Search Path Issues

   The DNS search path may vary for administrative reasons. For
   example, a site under the domain example.com may choose to place an
   early IPv6 deployment under the subdomain ipv6.example.com, until it
   is confident of offering a full dual-stack service under its main

   domain. The subtlety here is that the DNS search path then affects
   the choice of protocol used, such as IPv6 for nodes in
   ipv6.example.com.

3.6. Protocol Startup Sequence

   In the dual-stack environment, one needs to consider what happens if,
   for example, the IPv6 interface (transport) is started after DHCPv4
   was used to configure the client. Should the client then simply
   discard the current IPv4 information, or merge it with a subsequent
   IPv6 response? It may also be possible that one protocol is shut
   down or started while the system is running. There are similarities
   here to issues when DHCP renewals have information that may appear
   that previously was not available (or no longer carry information
   that has been removed).

3.7. DHCP Option Variations

   Some options in DHCP are not available in DHCPv6 and vice versa.
   Some IP-version limitations naturally apply; for example, only IPv6
   addresses can be in an IPv6 NTP option. The DHCP and DHCPv6 option
   numbers may be different.

   Some sites may choose to use IPv4-mapped addresses in DHCPv6-based
   options. The merits and drawbacks of such an approach need
   discussion.

   A site administrator may wish to configure all their dual-stack nodes
   with (say) two NTP servers, one of which has an IPv4 address, the
   other an IPv6 address. In this case, it may be desirable for an NTP
   option to carry a list of addresses, where some may be IPv4 and some
   may be IPv6. In general one could consider having DHCPv6 options
   that can carry a mix of IPv4 and IPv6 addresses.

3.8. Security Issues

   This document does not introduce any new security issues per se. A
   detailed analysis of DHCP and DHCPv6 security is out of scope for
   this document.

   While there is a specification for authentication for DHCP messages
   [3], the standard seems to have very few, if any, implementations.
   Thus DHCP and DHCPv6 servers are still liable to be spoofed. Adding
   an additional protocol may give an extra avenue for attack, should an
   attacker perhaps spoof a DHCPv6 server but not a DHCP server.

4. Potential Solutions

   Here we discuss the two broad solution strategies proposed within the
   IETF dhc WG. The first is to run separate DHCP and DHCPv6 servers
   (with the client merging information received from both where
   necessary, or perhaps choosing to query a particular version first).
   The second is to run only a DHCPv6 server and relay IPv4
   configuration information within (new) IPv4 configuration options.

4.1. Separate DHCP Servers

   One solution is to run separate DHCP and DHCPv6 servers. These may
   or may not be run on the same physical node. The information served
   from the DHCP servers could be generated from a single database
   instance for consistency. One might have a single server instance
   supporting both DHCPv4 and DHCPv6 protocols.

   In this approach, some best practice guidance is required for how
   multiple responses are handled or merged. Administrators have the
   onus to maintain consistency (for example, scripts may generate
   common DHCP and DHCPv6 configuration files).

   In some cases, inconsistencies may not matter. In a simple case, an
   NTP server will give the same time whether accessed by IPv4 or IPv6.
   Even if different recursive DNS servers are offered via DHCP or
   DHCPv6, then those name servers should provide the same response to a
   given query. In cases where sites may be operating a 'two-faced
   DNS', this will still hold true if the node is on the same
   topological point on the network from an IPv4 or IPv6 perspective.
   The order of DNS servers in a node's configuration is not important,
   unless DNS load balancing is required.

   In other cases, inconsistencies may be an issue; for example, where
   lists of values are returned, an algorithm is needed for list merger
   (e.g., "alternate, DHCPv6 first"). Or there may be incompatible
   configuration values where, for example, DHCPv6 supplies domain names
   (such the SMTP or POP servers) whereas DHCPv4 provides only IPv4
   addresses.

   In the case of separate servers, there are some options, like DNS
   search path, that aren't used in a specific IP protocol context.

   The multiple server approach will have some simplifications. The
   DHCPv4 and DHCPv6 servers may provide the same value for a particular
   parameter, in which case there is no conflict. In some cases, the
   value may be different, but the effect should be the same (such as an

NTP server). The crux of the issue is to identify where differences
may occur and where these differences will have an impact on node
behaviour.

One possible solution is to have per-host preferences, or an ordered
list of preferences, for example, "use manually configured", "prefer
DHCPv4", or "prefer DHCPv6", assuming the host can act based upon
which protocol is used. It is then up to the site administrator to
ensure that values returned from either DHCP are consistent (a
principle that extends if other methods are used, such as NIS or
Service Location Protocol (SLP)).

4.2. Single DHCPv6 Server

There is an argument for not having to configure and operate both
DHCP and DHCPv6 servers in a dual-stack site environment. The use of
both servers may also lead to some redundancy in the information
served. Thus, one solution may be to modify DHCPv6 to be able to
return IPv4 information. This solution is hinted at in the DHCPv6
[4] specification: "If there is sufficient interest and demand,
integration can be specified in a document that extends DHCPv6 to
carry IPv4 addresses and configuration information." This solution
may allow DHCP for IPv4 to be completely replaced by DHCPv6 with
additional IPv4 information options, for dual-stack nodes.

A general argument is that which DHCP protocol is used (whether it's
over IPv4 or IPv6) shouldn't affect what kind of addresses you can
get configured with it, and that simplicity and predictability come
from using a single server over a single transport. IPv4-capable
hosts will likely remain for at least 10 years, probably much longer;
do we want dual-stack hosts (which will become the norm) to do both
DHCPv4 and DHCPv6 forever while dual-stack? If you need both servers
to configure interfaces with addresses, and get other configurations,
then you rely on two separate protocols to work (servers and relays,
etc.) in order for the host to behave correctly.

This approach may require the listing of a mix of IPv4 and IPv6
addresses for an option. This could then be considered when new IPv6
options are introduced. There could be just two options needed, one
new option for the address delegation, and one for doing
encapsulation.

Also, there are a number of paradigms in DHCPv6 that we miss in
DHCPv4. An example is movement away from using MAC addresses for
per-host address assignment and instead using DHCP Unique Identifier
(DUIDs) or Identity Association Identifiers (IAIDs). As stated in
Section 9 of RFC3315, DHCPv6 servers use DUIDs to identify clients
for the selection of configuration parameters and in the association

of IAs with clients. DHCPv6 clients use DUIDs to identify a server
in messages where a server needs to be identified. However, in this
particular example, the new DHCPv6 functionality has recently been
retrofitted to IPv4 via a specification for DUIDs for DHCPv4 [6].

However, there are a number of potential problems with this approach:

o IPv4-only nodes would not have any DHCP service available to them;
   such an approach is only possible in a fully dual-stack
   environment.

o The client node may then be IPv6-only and receive IPv4
   configuration settings that it does not want or be able to handle
   meaningfully.

o The DHCPv4 servers need to be configured anyway to support IPv4-
   only hosts, so there is still duplication of information.

o What happens if there are DHCPv6 servers that don't return IPv4
   information? Does this mean the client can't run IPv4 (since it
   won't do DHCPv4)?

o If IPv4 information is served from a DHCPv6 server as well as an
   IPv4 DHCP server, IPv4 address space will need to be allocated to
   both servers, fragmenting the potentially precious IPv4 global
   address resource for the site.

4.3. Optimising for Failure with Lists of Addresses

   There is a generic issue with any option that includes a list of
   addresses of servers (such as DNS server addresses). The list is
   offered to cater for resilience, such as whether the listed server
   itself fails or connectivity to the server fails. If the client does
   not know the cause of failure, its optimal strategy is to try a
   different server, via a different protocol. The problem today is
   that the IPv4 list is returned via DHCPv4, and the IPv6 list via
   DHCPv6; the client really has no way to "try a different server",
   since that information is lost by the protocol, even though it may be
   known by the server.

   Just putting merging heuristics in the client cannot provide the best
   behaviour, since information is lost. By comparison, if IPv4-mapped
   addresses were included in the DHCPv6 option along with IPv6
   addresses, the DHCP server can give an intelligent order that takes
   into account which addresses are of the same DNS/whatever server.
   IPv6-only clients have to know to discard the IPv4-mapped addresses
   in this solution, and it's much easier to solve this in the combined-
   DHCP-server case than in the two-server case.

   One can argue that this is only an optimisation, and in many cases
   the list has only two elements, so the "next" choice is forced.
   However, this particular issue highlights the subtleties of merging
   responses from separate servers.

4.4. Administrative and Other Areas

   There are also administrative issues or best practice that could be
   promoted. For example, it may be recommended that sites do not split
   their DNS name space for IPv6-specific testbeds.

   It may be worth considering whether separate manual configuration
   files should be kept for IPv4 and IPv6 settings, such as separate
   /etc/resolv.conf files for DNS settings on UNIX systems. However,
   this seems a complex solution. The problem should be better solved
   by other, more generalised methods.

   It may be important at times to be able to distinguish DHCP client
   and server identities. DHCPv6 introduces the idea of a DHCP Unique
   Identifier (DUID). The DUID concept has also been retrofitted to
   DHCPv4 [6], and thus it may form the basis of part of the solution
   space for the problem at hand.

   Some differences in DHCP and DHCPv6 may not be reconciled, but may
   not need to be, such as different ways to assign addresses by DUID in
   DHCPv6, or the lack of a comparable option in both DHCP versions.

5. Summary

   There are a number of issues in the operation of DHCP and DHCPv6
   servers for nodes in dual-stack environments that should be
   clarified. While some differences in the protocols may not be
   reconciled, there may not be a need to do so. However, with DHCPv6
   deployment growing, there is an operational requirement to determine
   best practice for DHCP server provision in dual-stack environments,
   which may or may not imply additional protocol requirements. The
   principal choice is whether separate DHCP and DHCPv6 services should
   be maintained by a site, or whether DHCPv6 should be extended to
   carry IPv4 configuration settings for dual-stack nodes.

   It can certainly be argued that until a site is completely dual-
   stack, an IPv4 DHCP service will always be required (for example,
   while there are still legacy printers, IP webcams, or other devices
   that still configure via DHCPv4), and a single IPv6 transport DHCP
   server offering configuration information for both protocols will
   then not be sufficient. In that case, IPv4 DHCP is required, and
   thus there

is a good rationale for focusing effort on how to combine the
information received from separate IPv4 DHCP and (stateless) DHCPv6
servers.

In theory, it should be relatively straightforward to write a
configuration manager that would accept a single configuration
specification from the service manager and distribute the correct
(and consistent) configurations to the DHCPv4 and DHCPv6 servers
(whether on the same host or not). In this case, maintaining
coordinated configurations in two servers is an interface issue, not
a protocol issue. The question then is whether the client has all
the information it needs to make reasonable choices. We are aware of
one implementation of separate DHCPv4 and DHCPv6 clients that is
using a preference option for assisting client-side merging of the
received information.

Another issue for discussion is whether a combined DHCP service only
available over IPv6 transport is a desirable longer-term goal for
networks containing only dual-stack or IPv6-only nodes (or IPv4-only
nodes where DHCPv4 is not needed). The transition to the long-term
position may easily take more than 10 years.

Upon reflection on the above observations, the dhc WG reached a
strong consensus to adopt the two-server approach (separate DHCP and
DHCPv6 servers), rather than have a combined single server returning
IPv4 information over IPv6. The two servers may be co-located on a
single node and may have consistent configuration information
generated from a single asset database.

It should be noted that deployment experience of DHCPv6 is still in
its infancy; thus, a full understanding of the issues may only
develop over time, but we feel we have reached the best consensus
given the current status. Future work is now required to determine
best practice for merging information from multiple servers,
including merger of lists of addresses where options carry such
information.

As a footnote, we note that this work has overlap with multihoming
and multi-interface configuration issues. It is also interwoven with
the Detecting Network Attachment area, for example, where a node may
move from an IPv4-only network to a dual-stack network, or vice
versa. Both aspects may be best abstracted for discussion and
progression in the respective IETF multi6, shim6, and dna WGs, in
parallel with the two-server progression in the dhc WG.

6. Security Considerations

   There are no security considerations in this problem statement per
   se, as it does not propose a new protocol.

7. Acknowledgements

   The authors thank the following people for input to this document:
   Bernie Volz, AK Vijayabhaskar, Ted Lemon, Ralph Droms, Robert Elz,
   Changming Liu, Margaret Wasserman, Dave Thaler, Mark Hollinger, and
   Greg Daley. The document may not necessarily fully reflect the views
   of each of these individuals.

   The authors would also like to thank colleagues on the 6NET project
   for contributions to this document.

8. Informative References

   [1] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131,
        March 1997.

   [2] Thomson, S. and T. Narten, "IPv6 Stateless Address
        Autoconfiguration", RFC 2462, December 1998.

   [3] Droms, R. and W. Arbaugh, "Authentication for DHCP Messages",
        RFC 3118, June 2001.

   [4] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M.
        Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)",
        RFC 3315, July 2003.

   [5] Droms, R., "Stateless Dynamic Host Configuration Protocol (DHCP)
        Service for IPv6", RFC 3736, April 2004.

   [6] Lemon, T. and B. Sommerfeld, "Node-specific Client Identifiers
        for Dynamic Host Configuration Protocol Version Four (DHCPv4)",
        RFC 4361, February 2006.

Authors' Addresses

    Tim Chown
    University of Southampton
    School of Electronics and Computer Science
    Southampton, Hampshire   SO17 1BJ
    United Kingdom

    EMail: tjc@ecs.soton.ac.uk

    Stig Venaas
    UNINETT
    Trondheim   NO 7465
    Norway

    EMail: venaas@uninett.no

    Christian Strauf
    Clausthal University of Technology
    Erzstr. 51
    Clausthal-Zellerfeld   D-38678
    Germany

    EMail: strauf@rz.tu-clausthal.de

Full Copyright Statement

Intellectual Property

Acknowledgement