

Dynamic Host Configuration Working  
Group  
Internet-Draft  
Expires: April 27, 2006

S. Venaas  
T. Chown  
University of Southampton  
October 24, 2005

Dual-stack clients and merging of data from DHCPv4 and DHCPv6  
draft-ietf-dhc-dual-stack-merge-01

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with [Section 6 of BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at  
<http://www.ietf.org/ietf/lid-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at  
<http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on April 27, 2006.

Copyright Notice

Copyright (C) The Internet Society (2005).

Abstract

A node may have support for communications using both IPv4 and IPv6 protocols. Such a node may wish to obtain both IPv4 and IPv6 configuration settings via the Dynamic Host Configuration Protocol (DHCP). This can be done by using the IPv4 and the IPv6 DHC protocols respectively. This document considers mechanisms that allow such a node to make use of the configuration data from both protocols to obtain the desired common configuration.

Internet-Draft

Merging of DHCPv4 and DHCPv6 data

October 2005

## Table of Contents

<a href="#">1.</a>	Introduction . . . . .	<a href="#">3</a>
<a href="#">2.</a>	Tools for merging . . . . .	<a href="#">3</a>
<a href="#">2.1</a>	Host prefers IPv4 or IPv6 . . . . .	<a href="#">3</a>
<a href="#">2.2</a>	Dual-stack or both DHC protocols client option . . . . .	<a href="#">4</a>
<a href="#">2.3</a>	DUID and integrated DHCPv4/v6 server . . . . .	<a href="#">4</a>
<a href="#">2.4</a>	DHCPv6 option telling dual-stack client to use DHCPv4 . . . . .	<a href="#">4</a>
<a href="#">2.5</a>	IPv4-mapped addresses in DHCPv6 options . . . . .	<a href="#">4</a>
<a href="#">3.</a>	Solutions . . . . .	<a href="#">4</a>
<a href="#">3.1</a>	Use of preference rules . . . . .	<a href="#">4</a>
<a href="#">3.2</a>	Lists of mixed addresses . . . . .	<a href="#">5</a>
<a href="#">3.3</a>	Issues not solved . . . . .	<a href="#">6</a>
<a href="#">3.4</a>	Conclusion . . . . .	<a href="#">6</a>
<a href="#">4.</a>	IANA Considerations . . . . .	<a href="#">6</a>
<a href="#">5.</a>	Security Considerations . . . . .	<a href="#">6</a>
<a href="#">6.</a>	Informative References . . . . .	<a href="#">7</a>
	Authors' Addresses . . . . .	<a href="#">7</a>
	Intellectual Property and Copyright Statements . . . . .	<a href="#">8</a>

Internet-Draft

Merging of DHCPv4 and DHCPv6 data

October 2005

## [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 [\[2\]](#).

These protocols allow nodes to communicate via IPv4 or IPv6 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 [\[3\]](#), such a node may wish to use stateless DHCPv6 [\[4\]](#) 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. At the same time there may be IPv4-only and IPv6-only nodes using these protocols. Issues related to this have been described in [\[5\]](#). This document discusses approaches towards resolving these issues.

This initial revision does not attempt to describe any complete solutions, but rather serve as a discussion point by describing some of the possible methods that may be of use.

In this document, we refer to DHCP for IPv4 [\[1\]](#) as DHCPv4 and DHCP for IPv6 [\[2\]](#) as DHCPv6.

The authgors would welcome input on these approaches.

## [2.](#) Tools for merging

There are a number of different tools or methods that can be of use in ensuring that IPv4-only, IPv6-only and dual-stack hosts each get the information they need from DHCPv4, DHCPv6 or a combination of the

two.

### [2.1](#) Host prefers IPv4 or IPv6

The idea with a preference option of some kind is that a dual-stack host may obtain information from both DHCPv4 and DHCPv6 but will prefer one of them. So if a single valued option is received from both servers it can use the preferred one. For a set (or unordered list) it might use only the preferred result or mix them, while for an ordered list it should probably use all, but put the preferred first. The preference could be manually configured on the host or obtained via either DHCPv4 or DHCPv6. The option would only be

needed for one of them.

### [2.2](#) Dual-stack or both DHC protocols client option

A host could use a new DHCP option to tell the DHCP server (DHCPv4 or DHCPv6) that it is dual-stack and has or will request configuration for the other protocol. This can indicate to the server what information the server needs to return to the client.

### [2.3](#) DUID and integrated DHCPv4/v6 server

DHCPv6 [2] uses a DHCP Unique Identifier (DUID). A client requesting both IPv4 and IPv6, should use the same DUID for the two requests, as described in [6] for use of DUIDs with IPv4. If the client requests DHCPv4 first, then when it makes the DHCPv6 request, the server knows what information the client previously learnt through DHCPv4 from the observed DUID and could leave the duplicate information out from the DHCPv6 reply. We are not sure whether this can be done if multiple integrated servers are deployed, but it seems an interesting approach, and a good usage for DUIDs for IPv4.

### [2.4](#) DHCPv6 option telling dual-stack client to use DHCPv4

A new option could be used by a DHCPv6 server to tell a dual-stack client to request IPv4 information even if it has IPv4 addresses (tell client to use DHCPINFORM).

### [2.5](#) IPv4-mapped addresses in DHCPv6 options

DHCPv6 options could contain IPv4 addresses written as IPv4-mapped IPv6 addresses. This is not elegant, however.

### [3.](#) Solutions

We will now discuss how the above tools might be used to solve some of the issues in [\[5\]](#).

#### [3.1](#) Use of preference rules

A simple preference rule as in [Section 2.1](#) might be sufficient in many cases. The perhaps most difficult problem is where the option is a list of values, and one wishes to have a mix of IPv4 and IPv6 addresses where one does not want to list all of one IP type before the other, or if one is preferred to the other in most cases but not always. Lists of mixed addresses are discussed in [Section 3.2](#).

Another solution could be to use FQDNs as option values whenever possible. Then DHCPv4 and DHCPv6 might simply specify the same FQDN

where the FQDN is registered in the DNS with both IPv4 and IPv6 addresses. The preference would then be determined by the host's destination address selection rules. Some sites deploying IPv6 choose initially to use different FQDNs for IPv6, in which case this would not work.

The preference rule is not sufficient if say IPv6 is generally preferred, but IPv4 should be preferred in some cases. One way of doing this could be to have the client prefer IPv6 and make the DHCPv6 server omit IPv6 information for options where IPv4 is preferred. The server could do this if by use of the option in [Section 2.2](#) it knows that the client will also get the IPv4 information. An IPv6-only client, or one not requesting IPv4 configuration, should still get all the IPv6 options. The administrator may manually configure a DHCPv6 server to omit some of the IPv6 configuration for clients that also obtain IPv4 information. A combined DHCPv4 and DHCPv6 server might be able to determine this automatically. With different servers it might help to have a single combined admin interface.

One issue with the above is that the server must only omit options if it knows for sure that client will request and successfully obtain both IPv4 and IPv6 information. There are two ways this might be

done. One is that the server is told by the client that it uses both, by using the option in [Section 2.2](#), possibly combined with the option in [Section 2.4](#) where the server tells the client to request the other. Another possibly safer way is to make use of the DUID as in [Section 2.3](#) so that server knows that the client that previously made a DHCPv6 request, now makes a DHCPv4 request. The latter should work if a client generally preferring one protocol, uses DHCP for the preferred protocol last. We feel the DUID approach is an elegant one, and is a good use of the DUID concept that is now available for both DHCPv4 and DHCPv6.

### [3.2](#) Lists of mixed addresses

As we said previously, the most difficult problem is when one has a list of values, and one wishes to have a mix of IPv4 and IPv6 addresses where one does not want to list all of one IP type before the other. We are not sure if this is necessary to solve. If it is, the easiest solution might be to use IPv4-mapped addresses as in [Section 2.5](#) so that a mixed list of IPv4-mapped IPv6 addresses and other IPv6 addresses can be passed in a DHCPv6 option. If this is done it might be useful to have an option as described in [Section 2.2](#) that tells the server that the client is dual-stack. This is not elegant however, and one should certainly not pass mapped addresses to an IPv6-only host.

Another issue with using a simple preference for lists, is that if a

server is dual-stack with both IPv4 and IPv6 addresses, one may not wish to have both the addresses in the list. For example, if one has a nameserver with IPv4 address a4 and IPv6 address a6, and another with IPv4 address b4, one may not want the list "a6, a4, b4", but rather "a6, b4". Whether this is a problem may depend on whether the list is processed sequentially and how long timeout there is before trying the next in the list. If an integrated DHCPv4 and DHCPv6 server knows that a client has previously got the list "a6" via say DHCPv6, it could choose to omit "a4" when the same client makes a DHCPv4 query. It can detect that it is the same client using the DUID as in [Section 2.3](#). However if there are multiple integrated servers the two requests may go to different servers. Another alternative could be to use the option in [Section 2.2](#).

### [3.3](#) Issues not solved

There are many issues in [5] that are not tackled by the above. We have not looked at the issue of different people managing DHCPv4 and DHCPv6 or the case where the node is statically configured with information for one protocol while using DHCP for the other. Another issue is what to do when initially only one IP protocol is enabled, and the other is enabled later. There are other issues not sufficiently tackled as well, we suggest reading [5] for the full details. The methods presented here are just some preliminary ideas. Through discussion in the DHC WG we will try to come up with solutions that can resolve the issues. It may however not be possible to come up with a complete solution to all of them.

### [3.4](#) Conclusion

We have proposed some initial ideas for solving the issue of merging DHCP information available from DHCPv4 and DHCPv6 servers to IPv4-only, IPv6-only or dual-stack nodes. We would welcome feedback on these initial suggestions before progressing the document in more detail, and tackling the additional issues described in the problem statement draft. There are certainly useful tools for the task, in particular DUID identifiers now available for DHCPv4 and DHCPv6.

### [4.](#) IANA Considerations

This document has no actions for IANA.

### [5.](#) Security Considerations

We are not aware of any new security issues as a result of any of the described options, but this needs to be considered.

### [6.](#) Informative References

- [1] Droms, R., "Dynamic Host Configuration Protocol", [RFC 2131](#), March 1997.
- [2] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", [RFC 3315](#), July 2003.

- [3] Thomson, S. and T. Narten, "IPv6 Stateless Address Autoconfiguration", [RFC 2462](#), December 1998.
- [4] Droms, R., "Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6", [RFC 3736](#), April 2004.
- [5] Chown, T., "DHCP: IPv4 and IPv6 Dual-Stack Issues", [draft-ietf-dhc-dual-stack-03](#) (work in progress), July 2005.
- [6] Sommerfeld, B. and T. Lemon, "Node-Specific Client Identifiers for DHCPv4", [draft-ietf-dhc-3315id-for-v4-05](#) (work in progress), June 2005.

#### Authors' Addresses

Stig Venaas  
University of Southampton  
School of Electronics and Computer Science  
Southampton, Hampshire S017 1BJ  
United Kingdom

Email: [sv@ecs.soton.ac.uk](mailto:sv@ecs.soton.ac.uk)

Tim Chown  
University of Southampton  
School of Electronics and Computer Science  
Southampton, Hampshire S017 1BJ  
United Kingdom

Email: [tjc@ecs.soton.ac.uk](mailto:tjc@ecs.soton.ac.uk)

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at [ietf-ipr@ietf.org](mailto:ietf-ipr@ietf.org).

#### Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

#### Copyright Statement

Copyright (C) The Internet Society (2005). This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

#### Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.