

DHC Working Group  
INTERNET DRAFT  
Expires: July 2004  
Internet Draft  
Document: <[draft-lemon-dhcpv4-to-v6-id-trans-00.txt](#)>  
Category: Standards Track

Ted Lemon  
Nominum  
Bill Sommerfeld  
Sun Microsystems

January, 2004

Transition from [RFC2131](#)-style to [RFC3315](#)-style  
Client Identifiers for DHCPv4.

## Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#). 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/1id-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at:

<http://www.ietf.org/shadow.html>.

## Abstract

This document explores ways for a node that is configured using DHCP and that has in the past been configured using one of the client identification schemes described in [RFC2131](#) and [RFC2132](#) to make the transition to using the identification scheme described in [RFC3315](#) and in [draft-ietf-dhc-3315-id-for-v4.txt](#).

## 1. Problem Statement

The DHCPv4 protocol specification [[RFC2131](#), [RFC2132](#)] and the DHCPv6 protocol specification [[RFC3315](#)] both describe ways that DHCP clients can identify themselves to DHCP servers. Unfortunately, the methods described in these specifications are incompatible - a node that identifies itself according to the mechanism specified in [RFC3315](#) will use different identification information than a node that identifies itself according to [RFC2131](#)/[RFC2132](#), and it's not possible to describe a method for converting between the two types of identification.

The internet-draft, Node-Specific Client Identifiers for DHCPv4

[NODSPC], defines a new client identifier for DHCPv4 that is derived identically to the identifier used in DHCPv6. DHCPv4 clients that identify themselves using this method can have identifiers that are the same as the identifier sent by a DHCPv6 client running on the same node.

There is a problem with this, however. At the time that this document is being written, there are no DHCPv4 clients that use the mechanism described in [NODSPC]. In the case of IPv4-only nodes that will never run a DHCPv6 client, this is not a problem.

However, in the case where the node may be updated to run both IPv4 and IPv6, it may be useful for the DHCPv4 client to change client identifiers so that its client identifier is compatible with the one used by the DHCPv6 client. When it does this, any state on the DHCP server that is associated with the old client identifier will be lost. For example, the IP address formerly assigned to the client will not continue to be assigned to the client, and if the client has a domain name registered in the DNS, the client's association with that name will be lost.

An additional problem is that in administrative domains where long leases are assigned, when a client changes its client identifier, the server will wind up allocating it a second lease. If a large number of clients in a single administrative domain make the migration to new identifiers at the same time, this could result in address depletion. A more short-lived version of this problem can also happen in environments where DHCP servers implement per-customer lease limiting - because the lease limit per customer is probably quite small, when a customer attempts to migrate, there may not be an additional lease to allocate to the new client identifier until the lease associated with the old identifier is freed.

## [2.](#) Applicability

This draft proposes a number of solutions to the stated problem. The intention of the authors is that once one of these solutions is chosen, the draft should go to standards track.

## 3.. Proposed Solutions

### [3.1.](#) Do Nothing

One answer to this problem is to just switch client identifiers, but not do anything special to migrate resources associated with the client identifier. The way this will play out is as follows:

- The client will get a new IP address, if one is available.
- While the old lease is valid, any resources associated with it (e.g., the client's domain name) will be unavailable to the client.
- Once the old lease has expired, these resources (particularly the client's domain name) will be available for any client to claim.
- If a different client attempts to claim these resources first, that client will get them.

- If no other client attempts to claim the resources, the original client will reclaim them the first time it renews after the old lease has expired.

On networks where the client's IP address is not precious (for example, many NATted networks) and where the DHCP server does not maintain any other resources (e.g., domain names) on behalf of the client, this method is probably the best one, because it requires the least effort from the client and server.

### [3.2.](#) Propose New Identifier in DHCPREQUEST

Another answer to the problem is to require the client to send a DHCPREQUEST to transfer the resources to a new client identifier. The DHCPREQUEST message would send the old identifier in the client identifier option, and would send the proposed new identifier in a newly-defined option.

If the server supports the transition option, it sends back a DHCPACK with the new client identifier in the dhcp-client-identifier option. If it does not support the transition option, it either does not send back a client identifier option, or sends back a client identifier option that contains the old client identifier. [RFC2131](#) specifies the former behavior, but some existing DHCP server implementations send the client identifier anyway, so the client should be prepared for either possibility.

If the response indicates that the server doesn't support the transition option, the client sends a DHCPRELEASE to release the

lease and the resources associated with it, and then issues a DHCPDISCOVER using the new identifier to acquire a new lease. It uses the requested-address option to try to get the server to assign it the same lease. The DHCPRELEASE attempts to assure that any resources that are allocated to the client are released before the new identifier is used.

If, when the client is directed to change to the new identifier, it does not have a valid lease, it acquires a lease using the old client identifier and then follows the procedure described above.

Once a client that implements this method has made the transition one time, it always uses the new identifier in any subsequent DHCP messages, even if the server with which it is communicating changes.

This proposal works quite nicely for a DHCP client that is always in the same administrative domain. Such a client will never have more than one outstanding lease. This case coincides with the most likely case where the client is going to care about getting a consistent IP address. Clients that roam between administrative domains probably do not benefit very much either from having a stable IP address or from having the DHCP server maintain the

client's name in the DNS. So although this solution does not address all possible cases, it is nevertheless probably good enough.

### [3.3](#). Send Old Client Identifier

Another answer to the problem is for the client to send the new client identifier in the client-identifier option in any DHCP messages it sends after the transition has been made. In addition, it sends the old client identifier in a new option. When a compliant server looks up the client, it looks both under the old client identifier and under the new client identifier. If it finds the lease under the old identifier, it converts it to the new identifier. If it finds leases under both identifiers, the server uses the one that's associated with the new identifier and does nothing to the lease associated with the old identifier.

A client that implements this method needs to keep track of the last expiry time of a lease that was acquired under the old client identifier. Until that lease time has expired, it must continue to send the old client identifier option in every DHCP

message it sends. After that time has expired, it can forget that it ever used the old identifier.

This solution has the advantage that if a client has more than one outstanding lease recorded under the old identifier, it will potentially be able to reclaim all of those leases. One disadvantage is that leases held by non-compliant servers are never upgraded. Another disadvantage is that the client has to keep track of old leases, and it's possible that many existing clients do not keep track of this information, and thus would not be able to determine when it would be safe to stop sending the old client identifier.

#### 3.4. Probe for Leases Under Old Identifier

This is a more elaborate version of the solution described in [section 3.2](#). In this case, the client remembers all the leases it had. When it attaches to a new network segment, it sends a DHCPDISCOVER under the old client identifier, including the new client identifier option as described in [section 3.2](#). If it gets a DHCPOFFER listing one of the leases it remembers, it acquires that lease and then converts it using the method described in [section 3.2](#). It then removes the lease from its list of leases that need to be converted.

If the client doesn't recognize the lease it gets, it converts it anyway, as described in [section 3.2](#). Leases that are acquired during the probing process are never `_added_` to the list of leases needing to be converted.

If the server has a IP address associated with the new identifier, it sends that IP address to the client. In that case the client just starts using that lease.

When any lease on the list of leases to be converted expires, the client removes that lease from the list. When the list of leases to be converted is empty, the client no longer attempts to probe - at that point it is free to use the new client identifier, and need no longer remember that it made a transition from a different identifier.

This solution eliminates the problem with the solution proposed in [section 3.2](#), that only one lease can be converted. Thus, a roaming client can be sure that it will not run into problems. The downside to this proposal is that it required a great deal of

the client. The advantage is that it is completely compatible with the solution proposed in [section 3.2](#), which means that the implementor has the option of implementing either proposal.

#### [4.](#) Recommendations

The solutions proposed in sections [3.2](#) and [3.4](#) are compatible, and solve the stated problem nicely. So if a solution is to be implemented, it seems that pursuing one or both of these solutions would be the right thing to do. However, it's a lot of trouble to implement this, so it's worth discussing whether or not there's any advantage to undertaking this work, or whether it would be better not to try to solve this problem.

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

This draft introduces a new mechanism by which a malicious DHCP client could steal resources from an existing client. Currently, a malicious DHCP client that knows the client identifier of another client can send a DHCPRELEASE message to release the resources associated with that lease, and then send a DHCPDISCOVER message and attempt to acquire that client's lease. Using the mechanism proposed in sections [3.2](#) and [3.3](#), a malicious client could steal a lease in a single transaction, rather than using two transactions. It's not clear that this makes any difference - in order to avoid having this happen, the DHCP protocol needs to be protected with some kind of authentication scheme, for example the one defined in the DHCP Authentication specification [[RFC3118](#)]. It does not appear to be the case that that this proposal adds any new vulnerability.

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

This document may define a new DHCP option, and the code for that option would need to be assigned by the IANA.

#### [7.](#) Normative References

- [RFC2131] Droms, R., "Dynamic Host Configuration Protocol", [RFC 2131](#), March 1997.
- [[RFC2132](#)] S. Alexander, R. Droms, "DHCP Options and BOOTP Vendor Extensions", [RFC2132](#), March, 1997
- [[RFC3315](#)] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., Carney, M., "Dynamic Host Configuration Protocol for

IPv6 (DHCPV6)", July, 2003  
[NODSPC] Lemon, E., Sommerfeld, W., "Node-Specific Client  
Identifiers for DHCPv4", [draft-ietf-dhc-3315id-for-v4.txt](#),  
January, 2004

## 8. Informative References

[RFC3118] Droms, R., Arbaugh, W., "Authentication for DHCP  
Messages", [RFC3118](#), June, 2001

## Author's Addresses

Ted Lemon  
Nominum  
2385 Bay Road  
Redwood City, CA 94063 USA  
+1 650 381 6000  
[mellon@nominum.com](mailto:mellon@nominum.com)

Bill Sommerfeld  
Sun Microsystems  
1 Network Drive  
Burlington, MA 01824  
+1 781 442 3458  
[sommerfeld@sun.com](mailto:sommerfeld@sun.com)

## Full Copyright Statement

"Copyright (C) 2003, 2004 The Internet Society. All Rights Reserved.  
This document and translations of it may be copied and furnished to  
others, and derivative works that comment on or otherwise explain  
it or assist in its implementation may be prepared, copied,  
published and distributed, in whole or in part, without restriction  
of any kind, provided that the above copyright notice and this  
paragraph are included on all such copies and derivative  
works. However, this document itself may not be modified in any  
way, such as by removing the copyright notice or references to the  
Internet Society or other Internet organizations, except as needed  
for the purpose of developing Internet standards in which case the  
procedures for copyrights defined in the Internet Standards process  
must be followed, or as required to translate it into languages  
other than English.

The limited permissions granted above are perpetual and will not be

revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS 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."

#### Acknowledgement

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



