Network Working Group
INTERNET-DRAFT

Yakov Rekhter
Mark Stapp
Cisco Systems

February 1999
Expires August 1999

Interaction between DHCP and DNS
<draft-ietf-dhc-dhcp-dns-09.txt>

Status of this Memo

Copyright Notice

Abstract

DHCP provides a powerful mechanism for IP host autoconfiguration. However, the autoconfiguration provided by DHCP does not include updating DNS, and specifically updating the name to address and address to name mappings maintained by DNS.

This document specifies how DHCP clients and servers should use the Dynamic DNS Updates mechanism to update the DNS name to address and address to name mapping, so that the mappings for DHCP clients would be consistent with the IP addresses that the clients acquire via DHCP.

1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. Interaction between DHCP and DNS

DNS [RFC1034, RFC1035] maintains (among other things) the information about mapping between hosts' Fully Qualified Domain Names (FQDNs) [RFC1594] and IP addresses assigned to the hosts. The information is maintained in two types of Resource Records (RRs): A and PTR. The A RR contains mapping from a FQDN to an IP address; the PTR RR contains mapping from an IP address to a FQDN.

DHCP [RFC1541] provides a mechanism by which a host (a DHCP client) could acquire certain configuration information, and specifically its IP address(es). However, DHCP does not provide any mechanisms to update the DNS RRs that contain the information about mapping between the host's FQDN and its IP address(es) (A and PTR RRs). Thus the information maintained by DNS for a DHCP client may be incorrect - a host (the client) could acquire its address by using DHCP, but the A RR for the host's FQDN wouldn't reflect the address that the host acquired, and the PTR RR for the acquired address wouldn't reflect the host's FQDN.

Dynamic DNS Updates [RFC2136] is a mechanism that enables DNS information to be updated over a network.

The Dynamic DNS Update protocol can be used to maintain consistency between the information stored in the A and PTR RRs and the actual address assignment done via DHCP. When a host with a particular FQDN acquires its IP address via DHCP, the A RR associated with the host's FQDN would be updated (by using the Dynamic DNS Updates protocol) to reflect the new address. Likewise, when an IP address gets assigned to a host with a particular FQDN, the PTR RR associated with this address would be updated (using the Dynamic DNS Updates protocol) to reflect the new FQDN.

3. Models of operation

When a DHCP client acquires a new address, both the A RR (for the client's FQDN) and the PTR RR (for the acquired address) have to be updated. Therefore, we have two separate Dynamic DNS Update transactions. Acquiring an address via DHCP involves two entities: a DHCP client and a DHCP server. In principle each of these entities could perform none, one, or both of the transactions. However, upon some

reflection one could realize that not all permutations make sense. This document covers the possible design permutations:

(1) DHCP client updates the A RR, DHCP server updates the PTR RR

(2) DHCP server updates both the A and the PTR RRs

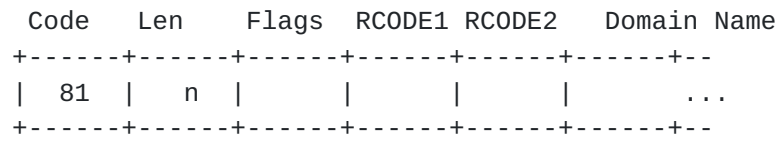
One could observe that the only difference between these two cases is whether the FQDN to IP address mapping is updated by a DHCP client or by a DHCP server. The IP address to FQDN mapping is updated by a DHCP server in both cases.

The reason these two are important, while others are unlikely, has to do with authority over the respective DNS RRs. A client may be given authority over mapping its own A RRs, or that may be restricted to a server to prevent the client from listing arbitrary addresses. In all cases, the only reasonable place for the authority over the PTR RRs associated with the address is in the DHCP server that allocates them.

3.1. Client FQDN Option

To update the IP address to FQDN mapping a DHCP server needs to know FQDN of the client to which the server leases the address. To allow the client to convey its FQDN to the server this document defines a new option, called "Client FQDN".

The code for this option is 81. Its minimum length is 4.



The Flags field allows a DHCP client to indicate to a DHCP server whether (a) the client wants to be responsible for updating the FQDN to IP address mapping (if Flags is set to 0), or (b) the client wants the server to be responsible for updating the FQDN to IP address mapping (if Flags is set to 1). The Flags field also allows a DHCP server to indicate to a DHCP client that the server assumes the responsibility for updating the FQDN to IP address mapping, even if the client wants to be responsible for this update (if Flags is set to 3).

The RCODE1 and RCODE2 fields are used by a DHCP server to indicate to a DHCP client the Response Code from Dynamic DNS Updates.

The Domain Name part of the option carries FQDN of a client.

3.2. DHCP Client behavior

The following describes behavior of a DHCP client that implements the Client FQDN option.

If a client that owns/maintains is own FQDN wants to be responsible for updating the FQDN to IP address mapping for the FQDN and address(es) used by the client, then the client MUST include the Client FQDN option in the DHCPREQUEST message originated by the client. The Flags field in the option MUST be set to 0. Once the client's DHCP configuration is completed (the client receives a DHCPACK message, and successfully completed a final check on the parameters passed in the message), the client MUST originate an update for the A RR (associated with the client's FQDN). The update MUST be originated following the procedures described in [RFC2136].

A client that owns/maintains its own FQDN can choose to delegate the responsibility for updating the FQDN to IP address mapping for the FQDN and address(es) used by the client to the server. In order to inform the server of this choice, the client MUST include the Client FQDN option in the DHCPREQUEST message originated by the client. The Flags field in the option MUST be set to 1. In this case, the client MAY supply an FQDN in the Client FQDN option, or it MAY leave that field empty as a signal to the server to generate an FQDN for the client in any manner the server chooses.

A client that delegates the responsibility for updating the FQDN to IP address mapping to a server MAY not receive any indications (either positive or negative) from the server whether the server was able to perform the update. In this case the client MAY use a DNS query to check whether the mapping is updated.

A client MUST set the RCODE1 and RCODE2 fields in the Client FQDN option to 0 when sending the option.

If a client releases its address lease prior to the lease expiration time and the client is responsible for updating its A RR(s), the client SHOULD delete the A RR (following the procedures described in [RFC2136]) associated with the leased address before sending a DHCP RELEASE message.

3.3. DHCP Server behavior

When a server receives a DHCPREQUEST message from a client, if the message contains the Client FQDN option, and the server replies to the message with a DHCPACK message, the server SHOULD originate an update for the PTR RR (associated with the address leased to the client). The update MUST be originated following the procedures described in Section 3.4. The server MAY complete the update before the server sends the DHCPACK message to the client. In this case the RCODE from the update [RFC2136] MUST be carried to the client in the RCODE1 field of the Client FQDN option in the DHCPACK message and the RCODE2 field MUST be set to 0. Alternatively, the server MAY send the DHCPACK message to the client without waiting for the update to be completed. In this case the RCODE1 field of the Client FQDN option in the DHCPACK message MUST be set to 255, and the RCODE2 field MUST be set to 0. The choice between the two alternatives is entirely up to the DHCP server.

In addition, if the Client FQDN option carried in the DHCPREQUEST message has its Flags field set to 1, then the server MUST originate an update for the A RR (associated with the FQDN carried in the option). The update MUST be originated following the procedures described in Section 3.4. The server MAY originate the update before the server sends the DHCPACK message to the client. In this case the RCODE from the update [RFC2136] MUST be carried to the client in the RCODE2 field of the Client FQDN option in the DHCPACK message. Alternatively the server MAY send the DHCPACK message to the client without waiting for the update to be completed. In this case the RCODE2 field of the Client FQDN option in the DHCKACK message MUST be set to 255. The choice between the two alternatives is entirely up to the DHCP server.

Even if the Client FQDN option carried in the DHCPREQUEST message has its Flags field set to 0 (indicating that the client wants to update the A RR), the server MAY (at the determination of the local administrator) update the A RR. The update MUST be originated following the procedures described in Section 3.4. The server MAY originate the update before the server sends the DHCPACK message to the client. In this case the RCODE from the update [RFC2136] MUST be carried to the client in the RCODE2 field of the Client FQDN option in the DHCPACK message, and the Flags field in the Client FQND option MUST be set to 3. Alternatively, the server MAY send the DHCPACK message to the client without waiting for the update to be completed. In this case the RCODE2 field of the Client FQDN option in the DHCKACK message MUST be set to 255, and the Flags field in the Client FQDN option MUST be set to 3. Whether the DNS update occurs before or after the DHCPACK is sent is entirely up to the DHCP server.

When a server receives a DHCPREQUEST message from a client, and the message contains the Client FQDN option, the server MUST ignore the value carried in the RCODE1 and RCODE2 fields of the option.

When a DHCP server sends the Client FQDN option to a client in the DHCPACK message, the server MUST copy the Domain Name fields from the Client FQDN option that the client sent to the server in the DHCPREQUEST message.

If the DHCPREQUST message received by a DHCP server from a DHCP client doesn't carry the Client FQDN option (e.g., the client doesn't implement the Client FQDN option), and the DHCP client acquires its FQDN from a DHCP server (as part of a normal DHCP transaction), then the server MAY be configured to update both A and PTR RRs. The updates MUST be originated following the procedures described in Section 3.4.

If a server detects that a lease on an address that the server leases to a client expires, the server SHOULD delete the PTR RR associated with the address. In addition, if the A RR (of the client) was initially updated by the server, the server SHOULD also delete the A RR. The deletion MUST follow the procedures described in [RFC2136].

If a server terminates a lease on an address prior to the lease expiration time, the server SHOULD delete the PTR RR associated with the address. In addition, if the server (that leased the address) initially updated the A RR (of the client), the server SHOULD also delete the A RR. The deletion MUST follow the procedures described in [RFC2136].

3.4. Procedures for performing DNS updates

There are two principal issues that need to be addressed concerning A RR DNS updates:

- Name Collisions

  If the entity updating the A RR (either the DHCP client or DHCP server) attempts to perform a DNS update with a DNS name that is already in use, what should be done? In some scenarios these name collisions are unlikely, in some scenarios they are very likely:

  1. Client updates A RR, uses DNSSEC: Name collisions in this scenario are unlikely (though not impossible), since for the client to use DNSSEC, it has already received credentials specific to the name it will add. This implies that the name has already been allocated (through some

implementation- or organization-specific procedure, and presumably uniquely) to that client.

2. Client updates A RR, uses some form of TSIG: Name collisions in this scenario are possible, since the credentials necessary for the client to update DNS are not name specific. Thus, for the client to be attempting to update a unique name requires the existence of some administrative procedure to ensure client configuration with unique names.

3. Server updates the A RR, uses a name for the client which is known to the server: Name collisions in this scenario are likely unless prevented by the server' name configuration procedures. See Section 7 for security issues with this form of deployment.

4. Server updates the A RR, uses a name supplied by the client: Name collisions in this scenario are highly likely, even with administrative procedures designed to prevent them. (This scenario is a popular one in real-world deployments in many types of organizations.) See Section 7 for security issues with this type of deployment.

Scenarios 3 and 4 are much more attractive given some form of DHCP Authentication, but the difficulties remain.

Scenarios 2, 3, and 4 rely on administrative procedures to ensure name uniqueness for DNS updates, and these procedures may break down. Experience has shown that, in fact, these procedures will break down at least occasionally. The question is what to do when these procedures break down or, for example in scenario #4, may not even exist.

In all cases of name collisions, the desire is to offer two modes of operation to the administrator of the combined DHCP-DNS capability: first-update-wins (i.e., the first updating entity gets the name) or most-recent-update-wins (i.e., the last updating entity for a name gets the name).

o Multiple DHCP servers

If multiple DHCP servers are able to update the same DNS zones, or if DHCP servers are performing A RR updates on behalf of DHCP clients, and more than one DHCP server may be able to serve addresses to the same DHCP clients, the DHCP servers should be able to provide reasonable DNS name update behavior for DHCP

clients.

The solution to both of these problems is for the updating entities (both DHCP clients or DHCP servers) to be able to cooperate when updating DNS A RRs.

Specifically, a KEY RR, described in [RFC2065] is used to associate client ownership information with a DNS name and the A RR associated with that name. When either a client or server adds an A RR for a client, it also adds a KEY RR which specifies a unique client identity (based on a "client specifier" created from the client's client-id or MAC address: see Appendix A). In this model, only one A RR is associated with a given DNS name at a time.

By associating this ownership information with each A RR, cooperating DNS updating entities may determine whether their client is the first or last updater of the name (and implement the appropriately configured administrative policy), and DHCP clients which currently have a host name may move from one DHCP server to another without losing their DNS name.

See Appendix A for the details of the use of the KEY RR for this purpose.

The specific algorithms utilizing the KEY RR to signal client ownership are explained below. The algorithms only work in the case where the updating entities all cooperate -- this approach is advisory only and does not substitute for DNS security, nor is it replaced by DNS security.

3.4.1. Adding A RRs to DNS

When a DHCP client or server intends to update an A RR, it first prepares a DNS UPDATE query which includes as a prerequisite the assertion that the name does not exist. The update section of the query attempts to add the new name and its IP address mapping and the KEY RR with its unique client-identity.

If this update operation succeeds, the updater can conclude that it has added a new name whose only RRs are the A and KEY RR records. The A RR update is now complete (and a client updater is finished, while a server would then proceed to perform a PTR RR update).

If the first update operation fails with YXDOMAIN, the updater can conclude that the intended name is in use. The updater then attempts to confirm that the DNS name is not being used by some other host. The updater prepares a second UPDATE query in which the prerequisite is that the desired name has attached to it a KEY RR whose contents

match the client identity (see Appendix A). The update section of this query deletes the existing A records on the name, and adds the A record that matches the DHCP binding and the KEY RR with the client identity.

If this query succeeds, the updater can conclude that the current client was the last user of this name, and that the name now contains the updated A RR. The A RR update is now complete (and a client updater is finished, while a server would then proceed to perform a PTR RR update).

If the second query fails with NXRRSET, the updater must conclude that the client's desired name is in use by another host. At this juncture, the updater can decide (based on some administrative configuration outside of the scope of this document) whether to let the existing owner of the name keep that name, and to (possibly) perform some name disambiguation operation on behalf of the current client, or to 'take-over' the name on behalf of the current client.

DISCUSSION:

The updating entity may be configured to allow the existing owner to keep the name, and to perform disambiguation on the name of the current client in order to attempt to generate a similar but unique name for the current client. In this case, once such a similar name has been generated, the updating entity should restart the process of adding an A RR as specified in this section.

3.4.2. 2 Adding PTR RR Entries to DNS

The DHCP server submits a DNS query which deletes all of the PTR RRs associated with the lease IP address, and adds a PTR RR whose data is the client's (possibly disambiguated) host name. The server also adds a KEY RR whose data is the client's client-identity as described in Appendix A.

3.4.3. Removing Entries from DNS

The first rule in removing DNS entries is be sure that an entity removing a DNS entry is only removing an entry that it added.

When a lease expires or a DHCP client issues a DHCPRELEASE request, the DHCP server SHOULD delete the PTR RR that matches the DHCP binding, if one was successfully added. The server's update query SHOULD assert that the name in the PTR record matches the name of the client whose lease has expired or been released.

The entity chosen to handle the A record for this client (either the

client or the server) SHOULD delete the A record that was added when the lease was made to the client.

In order to perform this delete, the updater prepares an UPDATE query which contains two prerequisites. The first prerequisite asserts that the KEY RR exists whose data is the client identity described in Appendix A. The second prerequisite asserts that the data in the A RR contains the IP address of the lease that has expired or been released.

If the query fails, the updater MUST conclude that it cannot delete the DNS name. It may be that the host whose lease on the server has expired has moved to another network and obtained a lease from a different server, which has caused the client's A RR to be replaced. It may also be that some other client has been configured with a name that matches the name of the DHCP client, and the policy was that the last client to specify the name would get the name. In this case, the KEY RR will no longer match the updater's notion of the client-identity of the host pointed to by the DNS name.

4. Updating other RRs

The procedures described in this document only cover updates to the A and PTR RRs. Updating other types of RRs is outside the scope of this document.

5. Security Considerations

Whether the client wants to be responsible for updating the FQDN to IP address mapping, or whether the client wants to delegate this responsibility to a server is a local to the client matter. The choice between the two alternatives may be based on a particular security model that is used with the Dynamic DNS Update protocol (e.g., only a client may have sufficient credentials to perform updates to the FQDN to IP address mapping for its FQDN).

Whether a DHCP server is always responsible for updating the FQDN to IP address mapping (in addition to updating the IP to FQDN mapping), regarless of the wishes of a DHCP client, is a local to the server matter. The choice between the two alternatives may be based on a particular security model.

The client SHOULD use some form of data origin authentication procedures (e.g., DNSSEC [DNSSEC]) when performing DNS updates.

While the DHCP client SHOULD be the one to update the DNS A record,

in certain specialized cases a DHCP server MAY do so instead. In this case, the DHCP server MUST be sure of both the name to use for the client, as well as the identity of the client.

In the general case, both of these conditions are not satisfied -- one needs to be mindful of the possibility of MAC address spoofing in a DHCP packet. It is not difficult for a DHCP server to know unambiguously the DNS name to use for a client, but only in certain relatively unusual circumstances will the DHCP server know for sure the identity of the client. One example of such a circumstance is where the DHCP client is connected to a network through an MCNS cable modem, and the CMTS (head-end) of the cable modem ensures that MAC address spoofing simply does not occur.

Another example where the DHCP server would know the identity of the client would be in a case where it was interacting with a remote access server which encoded a client identification into the DHCP client-id option. In this case, the remote access server as well as the DHCP server would be operating within a trusted environment, and the DHCP server could trust that the user authentication and authorization procedure of the remote access server was sufficient, and would therefore trust the client identification encoded within the DHCP client-id.

In either of these cases, a DHCP server would be able to correctly enter the DNS A record on behalf of a client, since it would know the name associated with a client (through some administrative procedure outside the scope of this protocol), and it would also know the client's identity in a secure manner.

6. Appendix A - Use of the KEY RR

The KEY RR used to hold the DHCP client's identity is formatted as follows:

The name of the KEY RR is the name of the A or PTR RR which refers to the client.

The flags field is set to 0x42 - that is, the 1 bit and the 6 bit are set.

The protocol field is set to 0.

The algorithm field is set to 254.

The first byte in the key field contains the length of the client-identity, and is followed by that number of bytes. If a DHCP client sent the client-id option in its request, the client-identity MUST be

identical to the data in the client-id option. If a client did not send the client-id option, the client-identity is constructed from the htype byte, the hlen byte, and hlen bytes of the client's chaddr from its request message.

7. References

[RFC1034] P. Mockapetris, "Domain names - concepts and facilities", RFC1034, 11/01/1987

[RFC1035] P. Mockapetris, "Domain names - implementation and specification", RFC1035, 11/01/1987

[RFC2131] R. Droms, "Dynamic Host Configuration Protocol", RFC2131, March 1997

[RFC1594] A. Marine, J. Reynolds, G. Malkin, "FYI on Questions and Answer Answers to Commonly asked ``New Internet User'' Questions", RFC1594, 03/11/1994

[DNSSEC]

[RFC2136] P. Vixie, S. Thomson, Y. Rekhter, J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)", RFC2136, April 1997

[RFC2119] Bradner, S. "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119.

[RFC2065] D. Eastlake, C. Kaufman, "Domain Name System Security Extensions", RFC 2065, January 1997.

8. Acknowledgements

Many thanks to Mark Beyer, Jim Bound, Ralph Droms, Peter Ford, Edie Gunter, Kim Kinnear, Stuart Kwan, Ted Lemon, Michael Lewis, Michael Patton, and Glenn Stump for their review and comments.

9. Author information

Yakov Rekhter
Cisco Systems, Inc.
170 Tasman Dr.
San Jose, CA 95134
Phone: (914) 235-2128
email: yakov@cisco.com

Mark Stapp
Cisco Systems
250 Apollo Drive
Chelmsford, MA 01824
Phone: (978) 244-8498
email: mjs@cisco.com

10. Full Copyright Statement