Internet Engineering Task Force B. Volz
INTERNET DRAFT Ericsson
DHC Working Group June 2002
Expires: December 18, 2002

Load Balancing for DHCPv6
draft-ietf-dhc-dhcpv6-loadb-01.txt

Status of this Memo

Copyright Notice

Abstract

This document specifies a load balancing algorithm for use with DHCPv6. Load balancing enables multiple cooperating DHCPv6 servers to decide which one should service a client, without exchanging any information beyond initial configuration. It expands on RFC 3074 "DHC Load Balancing Algorithm" to include DHCPv6.

1. Introduction

This document extends the load balancing concepts described in RFC 3074 "DHC Load Balancing Algorithm" [3] to DHCPv6 [2].

2. Requirements

The keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD,

SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL, when they appear in this document, are to be interpreted as described in RFC 2119 [1].

3. Terminology

This document uses terminology specific to IPv6 and DHCPv6 as defined in the "Terminology" section of the DHCP specification [2].

This document uses many of the concepts and terminology specific to load balancing as defined in the "Load Balancing Terminology" section of the DHC Load Balancing specification [3].

4. Motivation for Load Balancing

DHCP [2] provides for multiple servers to advertise service to the clients on links. A client is generally offered configuration service from each of the servers and there is no guarantee of consistency for the client (a different server may be selected each time).

Load balancing provides a quick and easy way for a server to determine whether it should service a particular client. Only the selected server or servers respond to the client instead of all of the servers. Load balancing provides a means to efficiently and consistently distribute the processing load for clients across multiple servers rather than having each server respond to every client.

In addition, rather than having multiple servers service the same clients, load balancing allows each server to service a different set of clients. If a server is down, the other servers may take over the clients that the downed server was to handle by monitoring the elapsed time option in client requests.

The load balancing technique described here and in RFC 3074 [3] work well for request/reply transaction protocols where a consistent client identifier is available.

For example, a high performance (non-redundant) configuration of DHCP servers might be as follows:

```
    +---------------+         +---------------+
    | DHCP Server 1 |         | DHCP Server 2 |
    |   HBA 0-127   |         |  HBA 128-255  |
    +-------+-------+         +-------+-------+
            |                         |
            |                         |
  <---------+-------------------------+-------Network--->
```

In this example, rather than both servers servicing all clients, each services appropriate half the clients and each services the same set of clients consistently. A redundant set of servers could be added (each configured with appropriate HBAs).

5. DHCPv6 Server Operation

DHCPv6 uses a DUID (DHCP Unique Identifier) to identify clients. The DUID is carried in most client-generated messages in the Client Identifier option as described in [2]. The client's DUID is defined to be the Service Transaction ID (STID) [3].

DHCPv6 uses two types of client messages, those that are directed to a specific server and those that are directed to all servers. The messages directed to a specific server contain a Server Identifier option as described in [2]. The messages directed to all servers do not include a Server Identifier option.

For the messages directed to a specific server, this load balancing algorithm does not apply and a server processes that client's request if the Server Identifier option's DUID of the request matches its own and discards all other requests.

For the messages directed to all servers, the load balancing algorithm MAY be used to limit the clients that a server services if the request contains a Client Identifier option. The server uses the hash algorithm described in [3] on the client's DUID (the STID) and uses the resulting hash value to determine if the client is within the server's configured hash bucket assignment (HBA) [3]. If the hash value is assigned to the server, the server MUST process the client request (other server policy may of course determine how the request is processed and whether a reply is sent to the client). If the hash value is not assigned to the server, the server SHOULD NOT process the request. The server MAY process the request if the elapsed time value in the Elapsed Time option of the request exceeds a configured value (the Service Delay or SD in [3]). How the SD is configured for a server is outside the scope of this document.

For client requests which do not contain a Client Identifier option, there is no STID and thus all servers process these requests.

A load balancing server would have the following processing flow for received client messages:

1. If the Server Identifier option is present in the message, process the message as per [2].

2. If no Client Identifier option is present in the message, process the message as per [2].

3. If the Client Identifier option's DUID is within the server's hash bucket assignment, process as per [2].

4. If the Elapsed Time option is present in the message and its

value exceeds the configured threshold, process as per [2].

5. Otherwise, do not process the message because load balancing dictates that another server should be processing the message.

The hash bucket assignments for each server must be configured and care must be taken to assign each hash bucket to at least one server. How the hash buckets are configured in servers is outside the scope of this document.

If a single hash bucket is assigned to multiple servers, the logic a client uses to select a server applies (just as if there were multiple servers for clients without load balancing). For example, each server can be configured with a different server preference value [2].

6. DHCPv6 Relay Agent Operation

Relay agents MAY be configured to relay client requests using load balancing. A load balancing relay agent must be configured with additional information as to the hash buckets assigned to each server, in a manner similar to that presented in [3]. Care must be taken to assure consistent information if both relay agents and servers are configured with load balancing information.

A relay agent would have the following processing flow for received client messages:

1. If no Client Identifier option is present in the client's message, relay the message to all configured servers regardless of hash bucket assignments.

2. Otherwise, use the hash algorithm described in [3] on the DUID in the Client Identifier option and relay the message to the server or servers assigned that hash bucket.

Relay agents MUST be configured to forward client requests to all of the DHCPv6 servers that may be part of a load balancing group.

Note: If relay agents are configured to do load balancing, the Elapsed Time option will be ineffective in allowing any server (not just the servers in the load balancing group) to respond to a client's request.

7. DHCPv6 Client Operation

DHCPv6 clients need not be aware that load balancing is in use by the servers. A client operates as described in [2].

Client operation with respect to load balancing is the same as client operation with multiple servers. If a server that was servicing a client becomes unavailable for some reason, the client will eventually time-out and communicate with all servers. When this happens, if there are multiple servers assigned to handle that client's hash bucket, one or more of these remaining servers will respond. If there are no other servers for that hash bucket, other servers may respond once the elapsed time value in the Elapsed Time option exceeds their configured SD.

If there is only one server (either for all clients or for some of the hash buckets), failure of that server will prevent clients from obtaining or extending the lifetimes of addresses. However, there is no difference whether load balancing is used or not.

8. Security Considerations

This proposal in and by itself provides no security, nor does it impact existing security. See [2] for further details as to DHCPv6 security issues.

Servers using load balancing are responsible for ensuring that if the contents of the HBA are transmitted over the network as part of the process of configuring any server, that message be secured against tampering, since tempering with the HBA could result in a denial of service for some or all clients.

9. Acknowledgements

Thanks to the DHC Working Group for their time and input into the specification starting at IETF-52. Thanks also to the following individuals for their comments and questions (in alphabetical order) Stefan Berg, Herold Fagerberg, Ted Lemon, Tony Lindstrom, Thomas Narten, Anders Strand, and Jack Wong.

References

[1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[2] Droms (ed.), R., Bound, J., Volz, Bernie, Lemon, Ted, Perkins, C., Carney, M., "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", draft-ietf-dhc-dhcpv6-26 (work in progress), June 2002.

[3] Volz, B., Gonczi, S., Lemon, T., Stevens, R., "DHC Load Balancing Algorithm", RFC 3074, February 2001.

Author's Address

Bernie Volz
Ericsson
959 Concord Street
Framingham, MA 01701
USA

Phone: +1 508 875 3162
EMail: bernie.volz@ericsson.com

Full Copyright Statement

Acknowledgement

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