Internet Engineering Task Force (IETF)
Request for Comments: 6853
BCP: 180
Category: Best Current Practice
ISSN: 2070-1721

J. Brzozowski
Comcast Cable Communications
J. Tremblay
Videotron G.P.
J. Chen
Time Warner Cable
T. Mrugalski
ISC

February 2013

# DHCPv6 Redundancy Deployment Considerations

## Abstract

This document provides information for those wishing to use DHCPv6 to support their deployment of IPv6. In particular, it discusses the provision of semi-redundant DHCPv6 services.

## Status of This Memo

This memo documents an Internet Best Current Practice.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on BCPs is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc6853.

## Copyright Notice

## Table of Contents

## 1. Introduction

Redundancy and high availability for many components of IPv6 infrastructure are desirable and, in some deployments, mandatory. Unfortunately, for DHCPv6 there is currently no standards-based failover or redundancy protocol. An interim solution is to provide semi-redundant services: this document specifies an architecture by which this can be achieved.

## 2. Scope and Assumptions

DHCPv6 redundancy may be useful in a wide range of scenarios. Although the architecture suggested in this document is able to be used in a wide range of networks, just two deployment environments are discussed here: service provider and enterprise network. All other scenarios may be generalized to one of these two cases.

In the rest of the document, the following assumptions are made with regards to the existing DHCPv6 infrastructure, regardless of the environment being considered:

1. At least two DHCPv6 servers provide a service to the same clients. (The architecture does not limit the number of servers, and more may be provided if required.)

2. The existing DHCPv6 servers will not directly communicate or interact with one another in the assignment of IPv6 addresses and the provision of configuration information to requesting clients.

3. DHCPv6 clients are instructed to run stateful DHCPv6 to request at least one IPv6 address. Configuration information and other options (such as a delegated IPv6 prefix) may also be requested as part of the stateful DHCPv6 operation.

4. Clients participating in DHCPv6 configuration have to properly handle the preference option, including the processing of ADVERTISE messages as required by [RFC3315].

5. A DHCPv6 server failure does not imply a failure of any other network service or protocol (e.g., TFTP servers). The redundancy of any additional services configured by means of DHCPv6 are outside the scope of this document. (For example, a single DHCPv6 server may configure multiple TFTP servers, with preference for each TFTP server, as specified in [RFC5970].)

While the techniques described in this document provide some aspects of redundancy, it should be noted that complete redundancy will not be available until a DHCPv6 failover protocol is standardized. The requirements for such a protocol are described in [FAILREQ].

## 2.1. Applicability to Prefix Delegation

The same approaches discussed in this document can potentially be applied to prefix delegation (PD) [RFC3633]. One obvious drawback of using a split prefix model for PD is that use of resources is doubled. It should be noted that such applicability remains theoretical and was not investigated thoroughly during work on this document. As such, the applicability of presented mechanisms to the prefix delegation is outside of the scope of this document.

## 3. Service Provider Deployment

The service provider model represents cases where the network and end-user devices may be administered by separate entities.

The DHCPv6 clients include cable modems, customer gateways or home routers, and end-user devices: these are collectively referred to as Customer Premises Equipment (CPE). In some cases hosts may be configured directly using the service provider DHCPv6 infrastructure; in others, configuration may be via an intermediate router that is being configured by the provider DHCPv6 infrastructure. Either way, the service provider DHCPv6 infrastructure may be semi-redundant.

In discussing this environment, additional assumptions to those listed in Section 2 have been made:

1. The service provider edge routers and access routers are IPv6 enabled when required. These routers are, for example, CMTS (Cable Modem Termination System) for cable or DSLAM/BRAS (Digital Subscriber Link Access Multiplexer / Broadband Remote Access Server) for DSL.

2. CPE devices are instructed to perform stateful DHCPv6 to request at least one IPv6 address, delegated prefix, and/or configuration information. CPE devices may also be instructed to use stateless DHCPv6 [RFC3736] to acquire configuration information only, a situation that assumes the IPv6 address and prefix information has been acquired using other means.

3. The primary application of this architecture is for native IPv6 services. (Use and applicability to transition mechanisms are out of scope for this document.)

4. The CPE devices must implement a stateful DHCPv6 client [RFC3315]. Support for DHCPv6 prefix delegation [RFC3633] or stateless DHCPv6 [RFC3736] may also be implemented.

## 4. Enterprise Deployment

The enterprise deployment environment covers cases where end-user devices are direct consumers of the configuration provided by the DHCP servers without any intermediate devices (as was the case with home routers used in the service provider environment). Although enterprise IPv6 environments quite often use or require DHCPv6 relay agents, the relays do not influence or process the configuration in any way and merely act as a transport mechanism.

The additional assumptions made for this model beyond those listed in Section 2 are:

1. DHCPv6 clients are hosts and are considered end nodes, i.e., they consume provided configuration and do not use it to provision other devices. Examples of such clients include desktop computers, laptops, printers, other typical office equipment, and some mobile devices.

2. The DHCPv6 clients generally do not require the assignment of an IPv6 prefix delegation, and as such they typically do not support DHCPv6 prefix delegation [RFC3633].

## 5. Protocol Requirements

Implementation of the architecture for semi-redundant DHCPv6 services using existing protocols requires the component DHCPv6 clients, relays, and servers to have certain capabilities. The following sections describe the requirements of such devices.

### 5.1. DHCPv6 Servers

This interim architecture requires the DHCPv6 servers that are [RFC3315] compliant and support the necessary options. Support for stateful DHCPv6 and the DHCPv6 preference option [RFC3315] is essential to the architecture. For deployment scenarios where IPv6 prefix delegation is needed, DHCPv6 servers must support DHCPv6 prefix delegation as defined by [RFC3633]. Furthermore, the DHCPv6 servers must support [RFC3736] if stateless DHCPv6 is used.

### 5.2. DHCPv6 Relays

DHCPv6 relay agents must be [RFC3315] compliant and must support the ability to relay DHCPv6 messages to more than one destination.

### 5.3. DHCPv6 Clients

DHCPv6 clients are required to be compliant with [RFC3315] and support the necessary options required to support the solution depending on the mode of operations and desired behavior:

- If prefix delegation is required, DHCPv6 clients must support DHCPv6 prefix delegation as defined in [RFC3633].

- Clients must support the acquisition of at least one IPv6 address and configuration information using stateful DHCPv6 as specified by [RFC3315].

- Stateless DHCPv6 [RFC3736] may also be supported.

- DHCPv6 clients must recognize and adhere to the processing of the advertised DHCPv6 preference option sent by the DHCPv6 servers.

## 6. Deployment Models

At the time of writing, a standards-based DHCPv6 redundancy protocol is not available. In the interim solution presented here, existing DHCPv6 server implementations are used as-is to provide best effort, semi-redundant DHCPv6 services. The behavior of these services will, in part, be governed by the configuration of each of the servers. Various aspects of the DHCPv6 protocol [RFC3315] are used to yield the desired behavior, although there is no inter-server or inter-process communication to coordinate DHCPv6 events and/or activities.

The solution does not impact DHCPv4, so DHCP services for both IPv4 and IPv6 may operate simultaneously on the same physical server(s) or may operate on different ones.

This section defines three semi-redundant models. Although /64 prefixes are used throughout the following sections as examples, other prefix lengths may be used as well.

### 6.1. Split Prefixes

In the split prefixes model, each DHCPv6 server is configured with a unique, non-overlapping pool derived from the /64 prefix deployed for use within an IPv6 network. For example, distributing an allocated /64 such as 2001:db8:1:1::/64 between two servers would require that it be split into two /65 pools, 2001:db8:1:1:0000::/65 and 2001:db8:1:1:8000::/65.

Both DHCPv6 servers are simultaneously active and operational, and each allocates IPv6 addresses from the corresponding pools per device class. The address allocation is governed largely through the use of the DHCPv6 preference option, so the server with the higher preference value is always preferred. Additional proprietary mechanisms can be used to further enforce the favoring of one DHCP server over another. An example of such a scenario is presented in Figure 1.

It is important to note that, over time, it is possible that bindings will be unevenly distributed amongst the DHCPv6 servers, and no one server will be authoritative for all of them.

As defined in [RFC3315], a DHCPv6 ADVERTISE message with a preference option of 255 is an indicator to a DHCPv6 client to immediately begin a client-initiated message exchange by transmitting a REQUEST message to the server that sent the ADVERTISE. Alternatively, a DHCPv6 ADVERTISE message with no preference option (or one with a value less than 255) is an indicator to the client that it must wait for subsequent ADVERTISE messages before choosing the server to which is responds, as described in Section 17.1.2 of [RFC3315].

In the event of a DHCPv6 server failure, it is desirable (but not essential) for a server other than the server that originally responded to be able to rebind the client's lease. Given the proposed architecture, the remaining active DHCPv6 server will have a different address pool configured, making it technically incorrect to rebind the client in its current state. Ultimately, the rebinding will fail and the client will acquire a new binding from the pool configured in the active server.

To reduce the possibility that a client or some other element on the network will experience a disruption in service or access to relevant binding data, shorter values for T1, T2, valid, and preferred lifetimes can be used. The values for the last three can be adjusted or configured to minimize service disruption. Ideally, setting them equal (or nearly equal) can be used to trigger a DHCPv6 client to reacquire the IPv6 address, prefix, and/or configuration information almost immediately after the rebinding fails. It is important to note, however, that shorter values will create an additional load on the DHCPv6 servers.

While using a split prefix configuration model, the dynamic updates to DNS [RFC2136] can be coordinated to ensure that the DNS is properly updated with the current binding information. Challenges arise with regards to the update of the PTR resource record for IPv6 addresses since the DNS information may need to be overwritten in a failure condition. The use of split prefixes enables the differentiation of bindings and binding timing to determine which represents the current state. This becomes particularly important when DHCPv6 Leasequery [RFC5007] and/or DHCPv6 Bulk Leasequery [RFC5460] are used to determine lease or binding state.

Finally, a benefit of this scheme is that the use of separate pools per DHCPv6 server makes failure conditions more obvious and detectable.

```
                 +----------+                 +-----------+
                 | Client 1 +-\             +--+ Server 1  |
                 +----------+  \            |  +-----------+
                                \           |
                                 \          |
                                  \         |
                 +----------+      \        |  +-----------+
                 | Client 2 +---------------+--| Server 2  |
                 +----------+      /        |  +-----------+
                       .          /         .
                       .         /          .
                       .        /           .
                 +----------+  /            .  +-----------+
                 | Client N +-/             .--| n+1 Server|
                 +----------+                  +-----------+

                 Server 1
                 ========
                 Prefix = 2001:db8:1:1::/64
                 Pool = 2001:db8:1:1:0000::/65
                 Preference = 255

                 Server 2
                 ========
                 Prefix = 2001:db8:1:1::/64
                 Pool = 2001:db8:1:1:8000::/65
                 Preference = 0

                 Server n+1
                 ==========
                 Prefix, pool, and preference would
                 vary based on prefix definition
```

Figure 1: Split prefixes approach

## 6.2. Multiple Unique Prefixes

In the multiple prefix model, each DHCPv6 server is configured with a unique, non-overlapping prefix. A /64 pool equal to the prefix is configured on each server. For example, the 2001:db8:1:1::/64 pool would be assigned to a single DHCPv6 server for allocation to clients equal to its parent prefix 2001:db8:1:1::/64. The second DHCPv6 server could use 2001:db8:1:5::/64 as both pool and prefix. This would be repeated for each active DHCP server. An example of this scenario is presented in Figure 2.

The major difference between the split prefixes approach and the multiple unique prefixes approach is that the latter does not require prefixes to be adjacent.  In fact, the split prefixes approach can be considered a special case of the multiple unique prefixes approach.

This approach uses a unique prefix and ultimately a single pool per DHCPv6 server with the corresponding prefixes configured for use in the network.  The corresponding network infrastructure must in turn be configured to use multiple prefixes on the interface(s) facing the DHCPv6 clients.  The configuration is similar on all the servers, but a different prefix and a different preference are used for each DHCPv6 server.

This approach drastically increases the rate of consumption of IPv6 prefixes and also yields operational and management challenges related to the underlying network since a significantly higher number of prefixes need to be configured and routed.  It also does not provide a clean migration path to the desired solution using a standards-based DHCPv6 redundancy or failover protocol (which, of course, has yet to be specified).

The use of multiple unique prefixes provides benefits related to dynamic updates to DNS similar to those referred to in Section 6.1. The use of multiple unique prefixes enables the differentiation of bindings and binding timing to determine which represents the current state.  This becomes particularly important when DHCPv6 Leasequery [RFC5007] and/or DHCPv6 Bulk Leasequery [RFC5460] are used to determine lease or binding state.  The use of separate prefixes and pools per DHCPv6 server makes failure conditions more obvious and detectable.

```
                 +----------+                 +-----------+
                 | Client 1 +-\            +--+ Server 1  |
                 +----------+  \           |  +-----------+
                                \          |
                                 \         |
                                  \        |
                 +----------+      \       |  +-----------+
                 | Client 2 +--------------+--| Server 2  |
                 +----------+      /       |  +-----------+
                       .          /        .
                       .         /         .
                       .        /          .
                 +----------+  /           .  +-----------+
                 | Client N +-/            .--| n+1 Server|
                 +----------+                 +-----------+

                 Server 1
                 ========
                 Prefix = 2001:db8:1:1::/64
                 Pool = 2001:db8:1:1::/64
                 Preference = 255

                 Server 2
                 ========
                 Prefix = 2001:db8:1:5::/64
                 Pool = 2001:db8:1:5::/64
                 Preference = 0

                 Server 3
                 ========
                 Prefix = 2001:db8:1:f::/64
                 Pool = 2001:db8:1:f::/64
                 Preference = [1..254]
```

Figure 2: Multiple unique prefix approach

### 6.3. Identical Prefixes

In the identical prefix model, each DHCPv6 server is configured with the same overlapping prefix and pool deployed for use within an IPv6 network. Distribution between two or more servers, for example, would require that the same /64 prefix and pool be configured on all DHCP servers. For instance, the 2001:db8:1:1::/64 pool would be assigned to all the DHCPv6 servers for allocation to clients derived from the 2001:db8:1:1::/64 prefix. This would be repeated for each active DHCP server. An example of such a scenario is presented in Figure 3.

This approach uses the same prefix, length, and pool definition across multiple DHCPv6 servers. All other configuration parameters remain the same, with the exception of the DHCPv6 preference. Such an approach conceivably eases the migration of DHCPv6 services to fully support a standards-based redundancy or failover protocol once such solution becomes available. Similar to the split prefix architecture described above, this approach does not place any additional addressing requirements on the network infrastructure.

The use of identical prefixes provides no benefit or advantage related to dynamic DNS updates, support of DHCPv6 Leasequery [RFC5007] or DHCPv6 Bulk Leasequery [RFC5460]. In this case, all DHCP servers will use the same prefix and pool configurations making it less obvious that a failure condition or event has occurred.

```
                 +----------+                 +-----------+
                 | Client 1 +-\             +--+ Server 1  |
                 +----------+  \            |  +-----------+
                                \           |
                                 \          |
                                  \         |
                 +----------+      \        |  +-----------+
                 | Client 2 +--------------+--| Server 2  |
                 +----------+      /        |  +-----------+
                       .          /         .
                       .         /          .
                       .        /           .
                 +----------+  /            .  +-----------+
                 | Client N +-/             .--| n+1 Server|
                 +----------+                 +-----------+

                 Server 1
                 ========
                 Prefix = 2001:db8:1:1::/64
                 Pool = 2001:db8:1:1::/64
                 Preference = 255

                 Server 2
                 ========
                 Prefix = 2001:db8:1:1::/64
                 Pool = 2001:db8:1:1::/64
                 Preference = 0

                 Server 3
                 ========
                 Prefix = 2001:db8:1:1::/64
                 Pool = 2001:db8:1:1::/64
                 Preference = [1..254]
```

Figure 3: Identical prefix approach

## 7. Challenges and Issues

The lack of interaction between DHCPv6 servers introduces a number of challenges related to the operations of the same service instances in a production environment. The following areas are of particular concern:

o In the identical prefixes scenario, both servers must follow the same address allocation procedure, i.e., they both must use the same algorithm and the same policy to determine which address is going to be assigned to a specific client. Otherwise, there is a distinct chance that each server will assign the same address to two different clients. It is expected that both servers will receive each incoming REQUEST message. Usually, no special action is required to achieve this as REQUEST messages are sent to a multicast address by clients. Relays are expected to forward incoming client messages to all servers. The client indicates the chosen server by including its DHCP Unique Identifier (DUID) in the Server-ID option. The chosen server assigns the address and other configuration options, while the other server discards the incoming request. In case of a failure of one server, the other server will assign the same address by following the same algorithm and the same policy.

- Interactions with DNS server(s) using dynamic update for the same address when one or more DHCPv6 servers have become unavailable. This specifically becomes a challenge when (or if) nodes that were initially granted a lease:

  1. Attempt to renew or rebind the lease originally granted, or

  2. Attempt to obtain a new lease

  The DHCID resource record [RFC4701] allows identification of the current owner of the specific DNS data that is the target of an update [RFC2136]. [RFC4704] specifies how DHCPv6 servers and/or clients may perform updates. [RFC4703] provides a way to solve conflicts between clients. Although [RFC4703] deals with most cases, it is still possible to leave abandoned resource records. Consider the following scenario: there are two independent servers, A and B. Server A assigns a lease to a client and updates the DNS with an AAAA record for the assigned address. When the client renews, server A is not available and server B assigns a different lease. The DNS is again updated, so now two AAAA resource records are present for the client: there is no indication as to which of the two leases is active. If server A never recovers, its information may never be removed (although it should be noted that this case is somewhat similar to that of a single server crashing and leaving abandoned resource records).

- Interactions with DHCPv6 servers to facilitate the acquisition of IPv6 lease data by way of the DHCPv6 Leasequery [RFC5007] or DHCPv6 Bulk Leasequery [RFC5460] protocols when one or more DHCPv6 servers have granted leases to DHCPv6 clients and later became unavailable. If the lease data is required and the granting server is unavailable, it will not be possible to obtain any information about leases granted until one of the following has taken place:

1. The granting DHCPv6 server becomes available with all lease information restored.

2. The client has renewed or rebound its lease against a different DHCPv6 server.

It is important to note that any exchange of available leases and synchronization between DHCPv6 servers is not possible until a redundancy or failover protocol is standardized or proprietary solutions become available.

## 8. Security Considerations

Additional security considerations are created through the use of this interim architecture beyond what has been cited in Section 23 of [RFC3315]. In particular, the dynamic DNS update using the models defined in this document allows for the possibility of not removing abandoned DNS records even when using the conflict resolution mechanism defined in [RFC4703]. However, this is no worse than a case where a single deployed server crashes and its lease database cannot be recovered.

When using the identical prefixes model, care must be taken to ensure that all servers use the same lease allocation procedure and are configured with the same policy. If this guidance is not followed, there is a risk of assignment of the same lease to two separate clients. In some cases, that situation can be recovered by using Duplicate Address Detection (Neighbor Discovery) and the DECLINE mechanism (DHCPv6).

## 9. Acknowledgements

The authors would like to thank Bernie Volz, Kim Kinnear, Ralph Droms, David Hankins, Chuck Anderson, Ted Lemon, Stephen Farrel, Pete McCann, Robert Sparks, Martin Stiemerling, Brian Haberman, and Barry Leiba for their input and review.

Special thanks to Stephen Morris for his numerous spelling, grammar corrections, and proofreading.

This work has been partially supported by Department of Computer Communications (a division of Gdansk University of Technology) and the National Centre for Research and Development (Poland) under the European Regional Development Fund, Grant No. POIG.01.01.02-00-045/09-00 (Future Internet Engineering Project).

## 10. References

### 10.1. Normative References

[RFC2136] Vixie, P., Thomson, S., Rekhter, Y., and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)", RFC 2136, April 1997.

[RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

[RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, December 2003.

[RFC3736] Droms, R., "Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6", RFC 3736, April 2004.

[RFC4701] Stapp, M., Lemon, T., and A. Gustafsson, "A DNS Resource Record (RR) for Encoding Dynamic Host Configuration Protocol (DHCP) Information (DHCID RR)", RFC 4701, October 2006.

[RFC4703] Stapp, M. and B. Volz, "Resolution of Fully Qualified Domain Name (FQDN) Conflicts among Dynamic Host Configuration Protocol (DHCP) Clients", RFC 4703, October 2006.

[RFC4704] Volz, B., "The Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Client Fully Qualified Domain Name (FQDN) Option", RFC 4704, October 2006.

[RFC5007] Brzozowski, J., Kinnear, K., Volz, B., and S. Zeng, "DHCPv6 Leasequery", RFC 5007, September 2007.

[RFC5460] Stapp, M., "DHCPv6 Bulk Leasequery", RFC 5460, February 2009.

[RFC5970] Huth, T., Freimann, J., Zimmer, V., and D. Thaler, "DHCPv6 Options for Network Boot", RFC 5970, September 2010.

### 10.2. Informative References

[FAILREQ] Mrugalski, T. and K. Kinnear, "DHCPv6 Failover Requirements", Work in Progress, September 2012.

## Authors' Addresses

John Jason Brzozowski
Comcast Cable Communications
1306 Goshen Parkway
West Chester, PA  19380
USA

Phone: +1-609-377-6594
EMail: john_brzozowski@cable.comcast.com

Jean-Francois Tremblay
Videotron G.P.
612 Saint-Jacques
Montreal, Quebec  H3C 4M8
Canada

EMail: jf@jftremblay.com

Jack Chen
Time Warner Cable
13820 Sunrise Valley Drive
Herndon, VA  20171
USA

EMail: jack.chen@twcable.com

Tomasz Mrugalski
Internet Systems Consortium, Inc.
950 Charter St.
Redwood City, CA  94063
USA

Phone: +1 650 423 1345
EMail: tomasz.mrugalski@gmail.com