

Network Working Group  
Internet-Draft

Updates: 6126bis (if approved)  
Intended status: Standards Track  
Expires: January 4, 2018

M. Boutier  
J. Chroboczek  
IRIF, University of Paris-Diderot  
July 03, 2017

Source-Specific Routing in Babel  
draft-boutier-babel-source-specific-03

Abstract

This document describes an extension to the Babel routing protocol to support source-specific routing.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on January 4, 2018.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. TODOs . . . . .	2
2. Introduction and background . . . . .	2
3. Data Structures . . . . .	3
3.1. The Source Table . . . . .	3
3.2. The Route Table . . . . .	3
3.3. The Table of Pending Requests . . . . .	3
4. Data Forwarding . . . . .	3
5. Protocol Operation . . . . .	5
5.1. Source-specific messages . . . . .	5
5.2. Route Acquisition . . . . .	5
5.3. Wildcard retractions (update) . . . . .	5
5.4. Wildcard requests . . . . .	6
6. Compatibility with the base protocol . . . . .	8
6.1. Loop-avoidance . . . . .	8
6.2. Starvation and Blackholes . . . . .	8
7. Protocol Encoding . . . . .	9
7.1. Source Prefix sub-TLV . . . . .	9
7.2. Source-specific Update . . . . .	9
7.3. Source-specific (Route) Request . . . . .	10
7.4. Source-Specific Seqno Request . . . . .	10
8. IANA Considerations . . . . .	10
9. Security considerations . . . . .	10
10. References . . . . .	10
10.1. Normative References . . . . .	10
10.2. Informative References . . . . .	11
Authors' Addresses . . . . .	11

## 1. TODOs

- o Source Prefix sub-TLV type: TBD
- o check references (Section) for BABEL in 6126bis
- o define wildcard Requests behaviour

## 2. Introduction and background

Source-specific routing (also known as Source Address Dependant Routing, SAD Routing or SADR) is an extension to traditional next-hop routing where packets are routed according to both their destination and their source address. This document describes the source-specific routing extension to the Babel routing protocol as defined in 6126bis [BABEL].

Background information about source-specific routing is provided in [SS-ROUTING].

### 3. Data Structures

This extension adds some data to the data structures maintained by a Babel node.

#### 3.1. The Source Table

Every Babel node maintains a source table, as described in [BABEL], Section 3.2.5. A source-specific Babel node extends this table with the following field:

- o the source prefix (sprefix, splen) specifying the source address of packets to which this entry applies.

If a source table entry has a zero length source prefix (splen equals to 0), then the entry is a non-source-specific entry, and is treated just like a source table entry defined by the original Babel protocol.

With this extension the route entry contains a source which itself contains a source prefix. These are two very different concepts, and should not be confused.

#### 3.2. The Route Table

Every Babel node maintains a route table, as described in [BABEL], Section 3.2.6. With this extension, this table is indexed by the 5-tuple (prefix, plen, source prefix, source plen, router-id) obtained from the associated source table entry.

If a route table entry has a zero length source prefix, then the entry is a non-source-specific entry, and is treated just like a route table entry defined by the original Babel protocol.

#### 3.3. The Table of Pending Requests

Every Babel node maintains a table of pending requests, as described in [BABEL], Section 3.2.7. A source-specific Babel node extends this table with the following entry:

- o the source prefix being requested.

### 4. Data Forwarding

In next-hop routing, if two routing table entries overlap, then one is necessarily more specific than the other; the "longest prefix rule" specifies that the most specific applicable routing table entry is chosen.

With source-specific routing, there might no longer be a most specific applicable prefix: two routing table entries might match a given packet without one necessarily being more specific than the other. Consider for example the following fragment of a routing table:

```
(2001:DB8:0:1::/64, ::/0, A)
```

```
(::/0, 2001:DB8:0:2::/64, B)
```

This specifies that all packets with destination in 2001:DB8:0:1::/64 are to be routed through A, while packets with a source in 2001:DB8:0:2::/64 are to be routed through B. A packet with source 2001:DB8:0:2::42 and destination 2001:DB8:0:1::57 matches both rules, although neither is more specific than the other. A choice is necessary, and unless the choice being made is the same on all routers in a routing domain, persistent routing loops may occur. More informations are available in [SS-ROUTING] Section IV.C.

A Babel implementation **MUST** choose routing table entries by using the so-called destination-first ordering, where a routing table entry R1 is preferred to a routing table entry R2 when either R1's destination prefix is more specific than R2's, or the destination prefixes are equal and R1's source prefix is more specific than R2's. (In more formal terms, routing table entries are compared using the lexicographic product of the destination prefix ordering by the source prefix ordering.)

In practice, this means that a source-specific Babel implementation must take care that any lower layer that performs packet forwarding obey this semantics. In particular:

- o If the lower layers implement the destination-first ordering, then the Babel implementation **MAY** use them directly;
- o If the lower layers can hold source-specific routes, but not with the right semantics, then the Babel implementation **MUST** disambiguate the routing table by using a suitable disambiguation algorithm (see [SS-ROUTING] Section V.B for such an algorithm);
- o If the lower layers cannot hold source-specific routes, then a Babel implementation **MUST** silently ignore (drop) any source-specific routes.

## 5. Protocol Operation

This extension does not fundamentally change the operation of the Babel protocol. We only describe the fundamental differences between the original protocol and the extension in this section. The other mechanisms described in [BABEL] (Section 3) are extended to pairs of (destination, source) prefixes instead of just (destination) prefixes.

### 5.1. Source-specific messages

Three messages are used to communicate informations on routes: Updates, Route Requests and Seqno Requests. With this extension, these messages carry an additionnal source prefix if (and only if) the corresponding route is source-specific. More formally, an Update, a Route Request and a Seqno Request MUST carry a source prefix if they concern a source-specific route (non-zero length source prefix) and MUST NOT carry a source prefix otherwise (zero length source prefix). A message which carries a source prefix is said to be source-specific.

### 5.2. Route Acquisition

When a non-source-specific Babel node receives a source-specific update, it silently ignores it.

TODO{On receipt of a source-specific update (id, prefix, source prefix, seqno, metric), a source-specific Babel node behaves as described in [BABEL] Section 3.5.4 though indexing entries by (neigh, id, prefix, source prefix).} When a source-specific Babel node receives a non-source-specific update, it MUST treat this update as carrying a zero length source prefix.

### 5.3. Wildcard retractions (update)

The original protocol defines a wildcard update with AE equals to 0 as being a wildcard retraction. A node receiving a wildcard retraction on an interface must consider that the sending node retracts all the routes it advertised on this interface.

Wildcard retractions are used when a node is about to leave the network. Thus, this extension does not define source-specific wildcard retraction, but extends wildcard retraction to apply also to source-specific routes. More formally, a wildcard update MUST NOT carry a source prefix, and a source-specific Babel node receiving a (legacy) wildcard update MUST retract all routes it learns from this node (including source-specific ones).

#### 5.4. Wildcard requests

TODO: behaviour to be defined.

##### 5.4.1. Proposal 1

The original Babel protocol states that when a node receives a wildcard route request, it SHOULD send a full routing table dump. This extension does not change this statement: a source-specific node SHOULD send a full routing table dump when receiving a wildcard request.

Source-specific wildcard requests does not exist: a wildcard request SHOULD NOT carry a source prefix.

##### 5.4.2. Proposal 2

We assume that a mandatory sub-TLV has a corresponding non-mandatory sub-TLV. This proposal is like Proposal 3 but instead of having multiple wildcard request TLVs, one for each kind of route understood, we use one wildcard request with sub-TLVs corresponding to the extension. To have a full routing table dump, a node sends a wildcard requests with a non-mandatory Source sub-TLV.

A source-specific node SHOULD always attach a non-mandatory Source sub-TLV to its wildcard requests.

This proposal has been rejected because it implies to share the space of non-mandatory and mandatory sub-TLVs.

##### 5.4.3. Proposal 3 (mentionned by Juliusz)

The Babel protocol provides the ability to request a full routing table dump by sending a "wildcard request", a route request with the AE field set to 0. As the original protocol has no source-specific routes, such a request may only concern non-source-specific routes. This extension does not modify the semantics of wildcard requests in that sense: a wildcard request prompts the receiver to send its non-source-specific routes only, and a Babel node SHOULD NOT send any source-specific updates in reply to a wildcard request.

To obtain a dump of the source-specific routes, a source-specific wildcard request MUST be used. A source-specific wildcard request is a wildcard request carrying a zero length source prefix.

When a node receives a source-specific wildcard request, it SHOULD send a dump of its routes which are source-specific "only". It SHOULD NOT send any non-source-specific routes in reply to a source-

specific wildcard request. It SHOULD NOT send any source-specific routes which are under the effect of a future extension. Such extension should detail how to handle the possible combinations.

In consequence, a node requiring a full routing table dump must send both a non-source-specific wildcard request and a source-specific wildcard request.

#### 5.4.4. Proposal 4 (mentionned by Juliusz)

Wildcard requests are deprecated. Either deprecate it in 6126bis, or say the following.

A node receiving a wildcard request SHOULD ignore it.

This proposal has been rejected because wildcard requests speeds up the convergence of the network on boot. This is considered important.

#### 5.4.5. Proposal 5 (mentionned by David)

By default, a vanilla wildcard request triggers a dump of all non-specific routes. We define a new non-mandatory sub-TLV on Route Requests called "Requested Route Types" that contains an array of all the types of routes this request is requesting.

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Type = TBD | Length | RR Type 1 | RR Type 2...
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

We also create a registry of Requested Route (RR) types, for example:

0 = Regular

1 = Source-Specific

2 = TOS-specific

etc.

A node receiving a Requested Route Types sub-TLV in a wildcard request SHOULD sends back a dump of all its routes corresponding to the requested types or to a combination of these types.

## 6. Compatibility with the base protocol

The protocol extension defined in this document is, to a great extent, interoperable with the base protocol defined in [BABEL] (and all its known extensions). More precisely, if non-source-specific routers and source-specific routers are mixed in a single routing domain, Babel's loop-avoidance properties are preserved, and, in particular, no persistent routing loops will occur.

However, this extension is not compatible with the Experimental Track's Babel Routing Protocol (RFC 6126). It requires the mandatory sub-TLV introduced in [BABEL]. Consequently, this extension **MUST NOT** be used with routers implementing RFC 6126, otherwise persistent routing loops may occur.

### 6.1. Loop-avoidance

The extension defined in this protocol uses a new Mandatory sub-TLV to carry the source prefix information. As discussed in Section 4.4 of [BABEL], this encoding ensures that non-source-specific routers will silently ignore the whole TLV, which is necessary to avoid persistent routing loops in hybrid networks.

Consider two nodes A and B, with A source-specific announcing a route to (D, S). Suppose that B merely ignores the source prefix information when it receives the update rather than ignoring the sub-TLV, and reannounces the route as D. This reannouncement reaches A, which treats it as (D, ::/0). Packets destined to D but not sourced in S will be forwarded by A to B, and by B to A, causing a persistent routing loop:

```

      (D,S)                (D)
      <--                  <--
----- A ----- B
      -->
      (D,::/0)

```

### 6.2. Starvation and Blackholes

In general, discarding source-specific routes by non-source-specific routers will cause route starvation. Intuitively, unless there are enough non-source-specific routes in the network, non-source-specific routers will suffer starvation, and discard packets for destinations that are only announced by source-specific routers.

A simple yet sufficient condition for avoiding starvation is to build a connected source-specific backbone that includes all of the edge



routers, and announce a (non-source-specific) default route towards the backbone.

## 7. Protocol Encoding

This extension defines a new sub-TLV used to carry a source prefix by the three following existing messages: Update, Route Request and Seqno Request.

### 7.1. Source Prefix sub-TLV

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|  Type = TBD   |      Length      |  Source Plen  |  Source Prefix...
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Fields:

Type           Set to TBD to indicate a Source Prefix sub-TLV.

Length        The length of the body, exclusive of the Type and Length fields.

Source Plen   The length of the advertised source prefix. This MUST NOT be 0.

Source Prefix   The source prefix being advertised. This field's size is (Source Plen)/8 rounded upwards.

The Source Prefix field's encoding (AE) is the same as the Prefix's. It is defined by the AE field of the corresponding TLV.

Note that this sub-TLV is a Mandatory sub-TLV. The whole TLV MUST be ignored if that TLV is not recognized as described in Section 4.4. Otherwise, routing loops may occur.

### 7.2. Source-specific Update

The source-specific Update is an Update TLV with a Source Prefix sub-TLV. It advertises or retracts source-specific routes in the same manner than routes with non-source-specific Updates (see [BABEL]). This TLV MUST NOT be attached to wildcard updates.

Contrary to the destination prefix, this extension does not compress the source prefix attached to Updates. The destination prefix uses compression as defined in [BABEL] for Updates with Mandatory extensions.

However, as defined in [BABEL] (Section 4.5), the compression is allowed for the destination prefix of source-specific routes. Legacy implementation will correctly update their parser state, while ignoring the whole TLV afterwards.

### 7.3. Source-specific (Route) Request

TODO: A source-specific Route Request prompts the receiver to send an update for a given pair of destination and source prefixes. It MUST NOT be used to request a full routing table dump. The Source Prefix sub-TLV of a wildcard source-specific Route Request (Request with AE equals to 0 and a Source Prefix sub-TLV) MIGHT be ignored: a receiver MIGHT reply by a full routing table dump.

### 7.4. Source-Specific Seqno Request

A source-specific Seqno Request is just like a Seqno Request for a source-specific route. It uses the same mechanisms described in [BABEL].

## 8. IANA Considerations

IANA is instructed to add the following entry to the "Babel sub-TLV Types" registry:

+-----+	-----+	-----+
Type	Name	Reference
+-----+	-----+	-----+
TBD	Source Prefix	(this document)
+-----+	-----+	-----+

## 9. Security considerations

The extension defined in this document adds a new sub-TLV to three TLVs already present in the original Babel protocol. It does not by itself change the security properties of the protocol.

## 10. References

### 10.1. Normative References

[BABEL] Chroboczek, J., "The Babel Routing Protocol", Internet Draft draft-ietf-babel-rfc6126bis-02, May 2017.

## 10.2. Informative References

### [SS-ROUTING]

Boutier, M. and J. Chroboczek, "Source-Specific Routing", August 2014.

In Proc. IFIP Networking 2015. A slightly earlier version is available online from <http://arxiv.org/pdf/1403.0445>.

### Authors' Addresses

Matthieu Boutier  
IRIF, University of Paris-Diderot  
Case 7014  
75205 Paris Cedex 13  
France

Email: [boutier@irif.fr](mailto:boutier@irif.fr)

Juliusz Chroboczek  
IRIF, University of Paris-Diderot  
Case 7014  
75205 Paris Cedex 13  
France

Email: [jch@irif.fr](mailto:jch@irif.fr)

Network Working Group  
Internet-Draft  
Intended status: Experimental  
Expires: January 4, 2018

J. Chroboczek  
IRIF, University of Paris-Diderot  
July 3, 2017

Homenet profile of the Babel routing protocol  
draft-ietf-homenet-babel-profile-02

Abstract

This document defines the subset of the Babel routing protocol [RFC6126] and its extensions that a Homenet router must implement, as well as the interactions between HNCP [RFC7788] and Babel.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on January 4, 2018.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
1.1. Background . . . . .	2
2. The Homenet profile of Babel . . . . .	3
2.1. Requirements . . . . .	3
2.2. Non-requirements . . . . .	4
3. Interactions between HNCP and Babel . . . . .	5
3.1. Requirements . . . . .	5
3.2. Non-requirements . . . . .	6
4. Security Considerations . . . . .	6
5. Acknowledgments . . . . .	7
6. References . . . . .	7
6.1. Normative References . . . . .	7
6.2. Informative References . . . . .	7
Author's Address . . . . .	8

## 1. Introduction

The core of the Homenet protocol suite consists of HNCP [RFC7788], a protocol used for flooding configuration information and assigning prefixes to links, combined with the Babel routing protocol [RFC6126]. Babel is an extensible, flexible and modular protocol: minimal implementations of Babel have been demonstrated that consist of a few hundred of lines of code, while the "large" implementation includes support for a number of extensions and consists of over ten thousand lines of C code.

This document consists of two parts. The first specifies the exact subset of the Babel protocol and its extensions that is required by an implementation of the Homenet protocol suite. The second specifies how HNCP interacts with Babel.

## 1.1. Background

The Babel routing protocol and its extensions are defined in a number of documents:

- o The body of RFC 6126 [RFC6126] defines the core, unextended protocol. It allows Babel's control data to be carried over either link-local IPv6 or IPv4, and in either case allows announcing both IPv4 and IPv6 routes. It leaves link cost estimation, metric computation and route selection to the implementation. Distinct implementations of core RFC 6126 Babel will interoperate and maintain a set of loop-free forwarding paths, but given conflicting metrics or route selection policies may give rise to persistent oscillations.

- o The informative Appendix A of RFC 6126 suggests a simple and easy to implement algorithm for cost and metric computation that has been found to work satisfactorily in a wide range of topologies.
- o While RFC 6126 does not provide an algorithm for route selection, its Section 3.6 suggests selecting the route with smallest metric with some hysteresis applied. An algorithm that has been found to work well in practice is described in Section III.E of [DELAY-BASED].
- o The extension mechanism for Babel is defined in RFC 7557 [RFC7557].
- o Four RFCs and Internet-Drafts define optional extensions to Babel: HMAC-based authentication [RFC7298], source-specific routing [BABEL-SS], radio interference aware routing [BABEL-Z], and delay-based routing [BABEL-RTT]. All of these extensions interoperate with the core protocol as well as with each other.

## 2. The Homenet profile of Babel

### 2.1. Requirements

REQ1: a Homenet implementation of Babel MUST encapsulate Babel control traffic in IPv6 packets sent to the IANA-assigned port 6696 and either the IANA-assigned multicast group ff02::1:6 or to a link-local unicast address.

Rationale: since Babel is able to carry both IPv4 and IPv6 routes over either IPv4 or IPv6, choosing the protocol used for carrying control traffic is a matter of preference. Since IPv6 has some features that make implementations somewhat simpler and more reliable (notably link-local addresses), we require carrying control data over IPv6.

REQ2: a Homenet implementation of Babel MUST implement the IPv6 subset of the protocol defined in the body of RFC 6126.

Rationale: support for IPv6 routing is an essential component of the Homenet architecture.

REQ3: a Homenet implementation of Babel SHOULD implement the IPv4 subset of the protocol defined in the body of RFC 6126. Use of other techniques for acquiring IPv4 connectivity (such as multiple layers of NAT) is strongly discouraged.

Rationale: support for IPv4 will remain necessary for years to come, and even in pure IPv6 deployments, including code for

supporting IPv4 has very little cost. Since HNCP makes it easy to assign distinct IPv4 prefixes to the links in a network, it is not necessary to resort to multiple layers of NAT, with all of its problems.

REQ4: a Homenet implementation of Babel MUST implement source-specific routing for IPv6, as defined in draft-boutier-babel-source-specific [BABEL-SS]. This implies that it MUST implement the extension mechanism defined in RFC 7557.

Rationale: source-specific routing is an essential component of the Homenet architecture. The extension mechanism is required by source-specific routing. Source-specific routing for IPv4 is not required, since HNCP arranges things so that a single non-specific IPv4 default route is announced (Section 6.5 of [RFC7788]).

REQ5: a Homenet implementation of Babel MUST use metrics that are of a similar magnitude to the values suggested in Appendix A of RFC 6126. In particular, it SHOULD assign costs that are no less than 256 to wireless links, and SHOULD assign costs between 32 and 196 to lossless wired links.

Rationale: if two implementations of Babel choose very different values for link costs, combining routers from different vendors will cause sub-optimal routing.

REQ6: a Homenet implementation of Babel SHOULD distinguish between wired and wireless links; if it is unable to determine whether a link is wired or wireless, it SHOULD make the worst-case hypothesis that the link is wireless. It SHOULD dynamically probe the quality of wireless links and derive a suitable metric from its quality estimation. The algorithm described in Appendix A of RFC 6126 MAY be used.

Rationale: support for wireless transit links is a "killer feature" of Homenet, something that is requested by our users and easy to explain to our bosses. In the absence of dynamically computed metrics, the routing protocol attempts to minimise the number of links crossed by a route, and therefore prefers long, lossy links to shorter, lossless ones. In wireless networks, "hop-count routing is worst-path routing".

## 2.2. Non-requirements

NR1: a Homenet implementation of Babel MAY perform route selection by applying hysteresis to route metrics, as suggested in Section 3.6 of RFC 6126 and described in detail in Section III.E of [BABEL-RTT]. However, it MAY simply pick the route with the smallest metric.

Rationale: hysteresis is only useful in congested and highly dynamic networks. In a typical home network, stable and uncongested, the feedback loop that hysteresis compensates for does not occur.

NR2: a Homenet implementation of Babel MAY include support for other extensions to the protocol, as long as they are known to interoperate with both the core protocol and source-specific routing.

Rationale: a number of extensions to the Babel routing protocol have been defined over the years; however, they are useful in fairly specific situations, such as routing over global-scale overlay networks [BABEL-RTT] or multi-hop wireless networks with multiple radio frequencies [BABEL-Z]. Hence, with the exception of source-specific routing, no extensions are required for Homenet.

### 3. Interactions between HNCP and Babel

The Homenet architecture cleanly separates between configuration, which is done by HNCP, and routing, which is done by Babel. While the coupling between the two protocols is deliberately kept to a minimum, some interactions are unavoidable.

All the interactions between HNCP and Babel consist of HNCP causing Babel to perform an announcement on its behalf (in particular, under no circumstances does Babel cause HNCP to perform an action). How this is realised is an implementation detail that is outside the scope of this document: while it could conceivably be done using a private communication channel between HNCP and Babel, existing implementations have HNCP install a route in the operating system's kernel which is later picked up by Babel using the existing redistribution mechanisms.

#### 3.1. Requirements

REQ7: if an HNCP node receives a DHCPv6 prefix delegation for prefix P and publishes an External-Connection TLV containing a Delegated-Prefix TLV with prefix P and no Prefix-Policy TLV, then it MUST announce a source-specific default route with source prefix P over Babel.

Rationale: source-specific routes are the main tool that Homenet uses to enable optimal routing in the presence of multiple IPv6 prefixes. External connections with non-trivial prefix policies are explicitly excluded from this requirement, since their exact behaviour is application-specific.



REQ8: if an HNCP node receives a DHCPv4 lease with an IPv4 address and wins the election for NAT gateway, then it MUST act as a NAT gateway and MUST announce a (non-specific) IPv4 default route over Babel.

Rationale: the Homenet architecture does not use source-specific routing for IPv4; instead, HNCP elects a single NAT gateway and publishes a single default route towards that gateway ([RFC7788] 7788 Section 6.5).

REQ9: if an HNCP node assigns a prefix P to an attached link and announces P in an Assigned-Prefix TLV, then it MUST announce a route towards P over Babel.

Rationale: prefixes assigned to links must be routable within the Homenet.

### 3.2. Non-requirements

NR3: an HNCP node that receives a DHCPv6 prefix delegation MAY announce a non-specific IPv6 default route over Babel in addition to the source-specific default route mandated by requirement REQ7.

Rationale: since the source-specific default route is more specific than the non-specific default route, the former will override the latter if all nodes implement source-specific routing. Announcing an additional non-specific route is allowed, since doing that causes no harm and might simplify operations in some circumstances, e.g. when interoperating with a routing protocol that does not support source-specific routing.

NR4: an HNCP node that receives a DHCPv4 lease with an IPv4 address and wins the election for NAT gateway SHOULD NOT announce a source-specific IPv4 default route.

Homenet does not require support for IPv4 source-specific routing. Announcing IPv4 source-specific routes will not cause routing pathologies (blackholes or routing loops), but it might cause packets sourced in different parts of the Homenet to follow different paths, with all the confusion that this entails.

## 4. Security Considerations

Both HNCP and Babel carry their control data in IPv6 packets with a link-local source address, and implementations are required to drop packets sent from a global address. Hence, they are only susceptible to attacks from a directly connected link on which the HNCP and Babel implementations are listening.

The security of a Homenet network relies on having a set of trusted "internal" links that are secured at a lower layer (either physically or at the link layer); HNCP and Babel packets are only accepted when they originate on these trusted links (see Section 5 of [RFC7788]). External, leaf and guest links are not trusted, and any HNCP or Babel packets that are received on such links are ignored.

If untrusted links are used for transit, which is NOT RECOMMENDED, and therefore need to carry HNCP and Babel traffic, then HNCP and Babel MUST be secured using an upper-layer security protocol. While both HNCP and Babel support cryptographic authentication, at the time of writing no protocol for autonomous configuration of HNCP and Babel security has been defined.

## 5. Acknowledgments

A number of people have helped with definining the requirements listed in this document. I am especially indebted to Markus Stenberg for his help.

## 6. References

### 6.1. Normative References

- [BABEL-SS] Boutier, M. and J. Chroboczek, "Source-Specific Routing in Babel", draft-boutier-babel-source-specific-01 (work in progress), January 2015.
- [RFC6126] Chroboczek, J., "The Babel Routing Protocol", RFC 6126, February 2011.
- [RFC7557] Chroboczek, J., "Extension Mechanism for the Babel Routing Protocol", RFC 7557, May 2015.
- [RFC7788] Stenberg, M., Barth, S., and P. Pfister, "Home Networking Control Protocol", RFC 7788, DOI 10.17487/RFC7788, April 2016.

### 6.2. Informative References

- [BABEL-RTT] Jonglez, B. and J. Chroboczek, "Delay-based Metric Extension for the Babel Routing Protocol", draft-jonglez-babel-rtt-extension-01 (work in progress), May 2015.

[BABEL-Z] Chroboczek, J., "Diversity Routing for the Babel Routing Protocol", draft-chroboczek-babel-diversity-routing-01 (work in progress), February 2016.

[DELAY-BASED] Jonglez, B. and J. Chroboczek, "A delay-based routing metric", March 2014.

Available online from <http://arxiv.org/abs/1403.3488>

[RFC7298] Ovsienko, D., "Babel Hashed Message Authentication Code (HMAC) Cryptographic Authentication", RFC 7298, July 2014.

Author's Address

Juliusz Chroboczek  
IRIF, University of Paris-Diderot  
Case 7014  
75205 Paris Cedex 13  
France

Email: [jch@irif.fr](mailto:jch@irif.fr)

Network Working Group  
Internet-Draft  
Updates: RFC7788 (if approved)  
Intended status: Standards Track  
Expires: January 4, 2018

P. Pfister  
Cisco Systems  
T. Lemon  
Nominum, Inc.  
July 3, 2017

Special Use Domain '.home.arpa'  
draft-ietf-homenet-dot-09

Abstract

This document specifies the behavior that is expected from the Domain Name System with regard to DNS queries for names ending with '.home.arpa.', and designates this domain as a special-use domain name. 'home.arpa' is designated for non-unique use in residential home networks. Home Networking Control Protocol (HNCP) is updated to use the '.home.arpa' domain instead of '.home'.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on January 4, 2018.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Requirements Language . . . . .	3
3. General Guidance . . . . .	3
4. Domain Name Reservation Considerations . . . . .	3
5. Updates to Home Networking Control Protocol . . . . .	5
6. Security Considerations . . . . .	5
7. Delegation of 'home.arpa' . . . . .	7
8. IANA Considerations . . . . .	7
9. Acknowledgments . . . . .	7
10. References . . . . .	7
10.1. Normative References . . . . .	7
10.2. Informative References . . . . .	8
Authors' Addresses . . . . .	8

## 1. Introduction

Users and devices within a home network (hereafter "homenet") require devices and services to be identified by names that are unique within the boundaries of the homenet [RFC7368]. The naming mechanism needs to function without configuration from the user. While it may be possible for a name to be delegated by an ISP, homenets must also function in the absence of such a delegation. A default name with a scope limited to each individual homenet needs to be used.

This document corrects an error in [RFC7788], replacing '.home' with '.home.arpa' as the default domain-name for homenets. '.home' had been selected as the most user-friendly option. However, there are existing uses of '.home' that may be in conflict with this use: evidence indicates that '.home' queries frequently leak out and reach the root name servers [ICANN1] [ICANN2].

In addition, it's necessary, for compatibility with DNSSEC (Section 6), that an unsigned delegation be present for the name. There is an existing process for allocating names under '.arpa' [RFC3172]. No such process is available for requesting a similar delegation in the root at the request of the IETF, which does not administer that zone. As a result, the use of '.home' is deprecated.

This document registers the domain '.home.arpa.' as a special-use domain name [RFC6761] and specifies the behavior that is expected from the Domain Name System with regard to DNS queries for names whose rightmost non-terminal labels are '.home.arpa.'. Queries for names ending with '.home.arpa.' are of local significance within the

scope of a homenet, meaning that identical queries will result in different results from one homenet to another. In other words, a name ending in '.home.arpa' is not globally unique.

Although this document makes specific reference to RFC7788, it is not intended that the use of '.home.arpa' be restricted solely to networks where HNCP is deployed; it is rather the case that '.home.arpa' is the correct domain for uses like the one described for '.home' in RFC7788: local name service in residential homenets.

## 2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

## 3. General Guidance

The domain name '.home.arpa.' is to be used for naming within residential homenets. Names ending with '.home.arpa.' reference a locally-served zone, the contents of which are unique only to a particular homenet, and are not globally unique. Such names refer to nodes and/or services that are located within a homenet (e.g., a printer, or a toaster).

DNS queries for names ending with '.home.arpa.' are resolved using local resolvers on the homenet. Such queries MUST NOT be recursively forwarded to servers outside the logical boundaries of the homenet.

Some service discovery user interfaces that are expected to be used on homenets conceal information such as domain names from end users. However, it is still expected that in some cases, users will need to see, remember, and even type, names ending with '.home.arpa.'. It is therefore desirable that users identify the domain and understand that using it expresses the intention to connect to a service that is specific to the homenet to which they are connected. Enforcing the fulfillment of this intention is out of scope for this document.

## 4. Domain Name Reservation Considerations

This section defines the behavior of systems involved in domain name resolution when resolving queries for names ending with '.home.arpa.' (as per [RFC6761]).

1. Users can use names ending with '.home.arpa.' just as they would use any other domain name. The '.home.arpa' name is chosen to be

readily recognized by users as signifying that the name is addressing a service on the homenet to which the user's device is connected.

2. Application software SHOULD NOT treat names ending in '.home.arpa' differently than other names. In particular, there is no basis for trusting names that are subdomains of '.home.arpa' (see Section 6).
3. Name resolution APIs and libraries MUST NOT recognize names that end in '.home.arpa.' as special and MUST NOT treat them differently. Name resolution APIs MUST send queries for such names to a recursive DNS server that is configured to be authoritative for the '.home.arpa' zone appropriate to the homenet. One or more IP addresses for recursive DNS servers will usually be supplied to the client through router advertisements or DHCP. If a host is configured to use a resolver other than one that is authoritative for the appropriate '.home.arpa' zone, the client may be unable to resolve, or may receive incorrect results for, names in sub domains of '.home.arpa'.
4. Unless configured otherwise, recursive resolvers and DNS proxies MUST behave as described in Locally Served Zones ([RFC6303] Section 3). Recursive resolvers that can be used in a homenet MUST be configurable with a delegation to an authoritative server for that particular homenet's instance of the domain '.home.arpa', and, when so configured, MUST NOT attempt to look up a delegation for '.home.arpa' in the public DNS. Of course, from an implementation standpoint it may be that a hybrid name server acts as a caching resolver or DNS proxy for non-local domains and as an authoritative server for '.home.arpa' and other locally served zones, responding directly to queries for subdomains of '.home.arpa' rather than using a delegation.
5. No special processing of '.home.arpa' is required for authoritative DNS server implementations. However, it is possible that an authoritative DNS server might attempt to validate the delegation for a zone before answering authoritatively for that zone. In this situation, it would find an invalid delegation, and would not answer authoritatively. A server that implements this sort of check MUST be configurable so that either it does not do this check for the 'home.arpa' domain, or it ignores the results of the check.
6. DNS server operators MAY configure an authoritative server for '.home.arpa' for use in homenets and other home networks. The operator for the DNS servers authoritative for '.home.arpa' in

the global DNS will configure any such servers as described in Section 7.

7. 'home.arpa' is a subdomain of the 'arpa' top-level domain, which is operated by IANA under the authority of the Internet Architecture Board according to the rules established in [RFC3172]. There are no other registrars for .arpa.

## 5. Updates to Home Networking Control Protocol

The final paragraph of Home Networking Control Protocol [RFC7788], section 8, is updated as follows:

OLD:

Names and unqualified zones are used in an HNCP network to provide naming and service discovery with local significance. A network-wide zone is appended to all single labels or unqualified zones in order to qualify them. ".home" is the default; however, an administrator MAY configure the announcement of a Domain-Name TLV (Section 10.6) for the network to use a different one. In case multiple are announced, the domain of the node with the greatest node identifier takes precedence.

NEW:

Names and unqualified zones are used in an HNCP network to provide naming and service discovery with local significance. A network-wide zone is appended to all single labels or unqualified zones in order to qualify them. '.home.arpa' is the default; however, an administrator MAY configure the announcement of a Domain-Name TLV (Section 10.6) for the network to use a different one. In case multiple are announced, the domain of the node with the greatest node identifier takes precedence.

The '.home.arpa' special-use name does not require a special resolution protocol. Names for which the rightmost two labels are '.home.arpa' are resolved using the DNS protocol [RFC1035].

## 6. Security Considerations

A DNS record that is returned as a response to a query for an FQDN in the domain '.home.arpa.' is expected to have local significance. It is expected to be returned by a server involved in name resolution for the homenet the device is connected in. However, such response MUST NOT be considered more trustworthy than would be a similar response for any other DNS query.



Because '.home.arpa' is not globally scoped and cannot be secured using DNSSEC based on the root domain's trust anchor, there is no way to tell, using a standard DNS query, in which homenet scope an answer belongs. Consequently, users may experience surprising results with such names when roaming to different homenets. To prevent this from happening, it may be useful for the resolver to identify different homenets on which it has resolved names, but this is out of scope for this document.

It is not possible to install a trust anchor for this zone in the '.arpa' zone. The reason for this is that in order to do so, it would be necessary to have the key-signing key for the zone ([RFC4034] Section 5). Since the zone is not globally unique, no one key would work.

An alternative would be to install a authenticated denial of existence ([RFC4033] Section 3.2). However, this assumes that validation is being done on a caching resolver that is aware of the special local meaning of '.home.arpa'. If a host stub resolver attempts to validate a name in '.home.arpa', an authenticated denial of existence of 'home' as a subdomain of 'arpa.' would cause the validation to fail. Therefore, the only delegation that will allow names under '.home.arpa' to be resolved is an unsigned delegation.

Consequently, unless a trust anchor for the particular instance of the '.home.arpa' zone being validated is manually configured on the validating resolver, DNSSEC signing of names within the '.home.arpa' zone is not possible.

Although in principle it might be useful to install a trust anchor for a particular instance of '.home.arpa', it's reasonable to expect that a host with such a trust anchor might from time to time connect to more than one network with its own instance of '.home.arpa'. Such a host would be unable to access services on any instance of '.home.arpa' other than the one for which a trust anchor was configured.

It is in principle possible to attach an identifier to an instance of '.home.arpa' that could be used to identify which trust anchor to rely on for validating names in that particular instance. However, the security implications of this are complicated, and such a mechanism, as well as a discussion of those implications, is out of scope for this document.

## 7. Delegation of 'home.arpa'

In order to be fully functional, there must be a delegation of 'home.arpa' in the '.arpa' zone [RFC3172]. This delegation MUST NOT be signed, MUST NOT include a DS record, and MUST point to one or more black hole servers, for example BLACKHOLE-1.IANA.ORG and BLACKHOLE-2.IANA.ORG. The reason that this delegation must not be signed is that not signing the delegation breaks the DNSSEC chain of trust, which prevents a validating stub resolver from rejecting names published under 'home.arpa' on a homenet name server.

## 8. IANA Considerations

IANA is requested to record the domain name '.home.arpa' in the Special-Use Domain Names registry [SUDN]. IANA is requested, with the approval of IAB, to implement the delegation requested in Section 7.

## 9. Acknowledgments

The authors would like to thank Stuart Cheshire for his prior work on '.home', as well as the homenet chairs: Mark Townsley and Ray Bellis. We would also like to thank Paul Hoffman for providing review and comments on the IANA considerations section and Suzanne Woolf and Ray Bellis for their detailed review comments.

## 10. References

### 10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC3172] Huston, G., Ed., "Management Guidelines & Operational Requirements for the Address and Routing Parameter Area Domain ("arpa")", BCP 52, RFC 3172, DOI 10.17487/RFC3172, September 2001, <<http://www.rfc-editor.org/info/rfc3172>>.
- [RFC6303] Andrews, M., "Locally Served DNS Zones", BCP 163, RFC 6303, DOI 10.17487/RFC6303, July 2011, <<http://www.rfc-editor.org/info/rfc6303>>.
- [RFC6761] Cheshire, S. and M. Krochmal, "Special-Use Domain Names", RFC 6761, DOI 10.17487/RFC6761, February 2013, <<http://www.rfc-editor.org/info/rfc6761>>.

- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<http://www.rfc-editor.org/info/rfc8174>>.

## 10.2. Informative References

- [ICANN1] "New gTLD Collision Risk Mitigation", October 2013, <<https://www.icann.org/en/system/files/files/new-gtld-collision-mitigation-05aug13-en.pdf>>.
- [ICANN2] "New gTLD Collision Occurrence Management", October 2013, <<https://www.icann.org/en/system/files/files/resolutions-new-gtld-annex-1-07oct13-en.pdf>>.
- [RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <<http://www.rfc-editor.org/info/rfc1035>>.
- [RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, DOI 10.17487/RFC4033, March 2005, <<http://www.rfc-editor.org/info/rfc4033>>.
- [RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, DOI 10.17487/RFC4034, March 2005, <<http://www.rfc-editor.org/info/rfc4034>>.
- [RFC7368] Chown, T., Ed., Arkko, J., Brandt, A., Troan, O., and J. Weil, "IPv6 Home Networking Architecture Principles", RFC 7368, DOI 10.17487/RFC7368, October 2014, <<http://www.rfc-editor.org/info/rfc7368>>.
- [RFC7788] Stenberg, M., Barth, S., and P. Pfister, "Home Networking Control Protocol", RFC 7788, DOI 10.17487/RFC7788, April 2016, <<http://www.rfc-editor.org/info/rfc7788>>.
- [SUDN] "Special-Use Domain Names Registry", July 2012, <<http://www.iana.org/assignments/special-use-domain-names/special-use-domain-names.xhtml>>.

## Authors' Addresses

Pierre Pfister  
Cisco Systems  
Paris  
France

Email: pierre.pfister@darou.fr

Ted Lemon  
Nominum, Inc.  
800 Bridge Parkway  
Redwood City, California 94065  
United States of America

Phone: +1 650 381 6000  
Email: ted.lemon@nominum.com

HOMENET  
Internet-Draft  
Intended status: Standards Track  
Expires: February 16, 2017

D. Migault (Ed)  
Ericsson  
R. Weber  
Nominum  
R. Hunter  
Globis Consulting BV  
C. Griffiths

W. Cloetens  
SoftAtHome  
August 15, 2016

Outsourcing Home Network Authoritative Naming Service  
draft-ietf-homenet-front-end-naming-delegation-05.txt

## Abstract

Designation of services and devices of a home network is not user friendly, and mechanisms should enable a user to designate services and devices inside a home network using names.

In order to enable internal communications while the home network experiments Internet connectivity shortage, the naming service should be hosted on a device inside the home network. On the other hand, home networks devices have not been designed to handle heavy loads. As a result, hosting the naming service on such home network device, visible on the Internet exposes this device to resource exhaustion and other attacks, which could make the home network unreachable, and most probably would also affect the internal communications of the home network.

As result, home networks may prefer not serving the naming service for the Internet, but instead prefer outsourcing it to a third party. This document describes a mechanisms that enables the Home Network Authority (HNA) to outsource the naming service to the Outsourcing Infrastructure.

## Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on February 16, 2017.

#### Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

#### Table of Contents

1. Requirements notation . . . . .	3
2. Introduction . . . . .	3
3. Terminology . . . . .	4
4. Architecture Description . . . . .	6
4.1. Architecture Overview . . . . .	6
4.2. Example: Homenet Zone . . . . .	8
4.3. Example: HNA necessary parameters for outsourcing . . . . .	10
5. Synchronization between HNA and the Synchronization Server . . . . .	11
5.1. Synchronization with a Hidden Primary . . . . .	11
5.2. Securing Synchronization . . . . .	12
5.3. HNA Security Policies . . . . .	14
6. DNSSEC compliant Homenet Architecture . . . . .	14
6.1. Zone Signing . . . . .	14
6.2. Secure Delegation . . . . .	16
7. Handling Different Views . . . . .	17
7.1. Misleading Reasons for Local Scope DNS Zone . . . . .	17
7.2. Consequences . . . . .	18
7.3. Guidance and Recommendations . . . . .	18
8. Homenet Reverse Zone . . . . .	19
9. Renumbering . . . . .	19
9.1. Hidden Primary . . . . .	20
9.2. Synchronization Server . . . . .	21
10. Privacy Considerations . . . . .	22
11. Security Considerations . . . . .	22

11.1.	Names are less secure than IP addresses . . . . .	22
11.2.	Names are less volatile than IP addresses . . . . .	23
11.3.	DNS Reflection Attacks . . . . .	23
11.3.1.	Reflection Attack involving the Hidden Primary . . .	23
11.3.2.	Reflection Attacks involving the Synchronization Server . . . . .	25
11.3.3.	Reflection Attacks involving the Public Authoritative Servers . . . . .	26
11.4.	Flooding Attack . . . . .	26
11.5.	Replay Attack . . . . .	26
12.	IANA Considerations . . . . .	27
13.	Acknowledgment . . . . .	27
14.	References . . . . .	27
14.1.	Normative References . . . . .	27
14.2.	Informational References . . . . .	30
Appendix A.	Document Change Log . . . . .	31
Authors' Addresses	. . . . .	33

## 1. Requirements notation

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 [RFC2119].

## 2. Introduction

IPv6 provides global end to end IP reachability. End users prefer to use names instead of long and complex IPv6 addresses when accessing services hosted in the home network.

Customer Edge Routers and other Customer Premises Equipment (CPEs) are already providing IPv6 connectivity to the home network, and generally provide IPv6 addresses or prefixes to the nodes of the home network. In addition, [RFC7368] recommends that home networks be resilient to connectivity disruption from the ISP. This could be achieved by a dedicated device inside the home network that builds, serves or manage the Homenet Zone, thus providing bindings between names and IP addresses.

CPEs are of course good candidates to manage the binding between names and IP addresses of nodes. However, this could also be performed by another device in the home network that is not a CPE. In addition, a given home network may have multiple nodes that may implement this functionality. Since management of the Homenet Zone involves DNS specific mechanisms that cannot be distributed (primary server), when multiple nodes can potentially manage the Homenet Zone, a single node needs to be selected. This selected node is designated as the Homenet Naming Authority (HNA).

CPEs, Homenet Naming Authority, as well as home network devices are usually low powered devices not designed not for terminating heavy traffic. As a result, hosting an authoritative DNS service on the Internet may expose the home network to resource exhaustion and other attacks. This may isolate the home network from the Internet and also impact the services hosted by the such an home network device, thus affecting overall home network communication.

In order to avoid resource exhaustion and other attacks, this document describes an architecture that outsources the authoritative naming service of the home network. More specifically, the Homenet Naming Authority builds the Homenet Zone and outsources it to an Outsourcing Infrastructure. The Outsourcing Infrastructure is in charge of publishing the corresponding Public Homenet Zone on the Internet.

Section 4.1 provides an architecture description that describes the relation between the Homenet Naming Authority and the Outsourcing Architecture. In order to keep the Public Homenet Zone up-to-date Section 5 describes how the Homenet Zone and the Public Homenet Zone can be synchronized. The proposed architecture aims at deploying DNSSEC, and the Public Homenet Zone is expected to be signed with a secure delegation. The zone signing and secure delegation may be performed either by the Homenet Naming Authority or by the Outsourcing Infrastructure. Section 6 discusses these two alternatives. Section 7 discusses the consequences of publishing multiple representations of the same zone also commonly designated as views. This section provides guidance to limit the risks associated with multiple views. Section 8 discusses management of the reverse zone. Section 9 discusses how renumbering should be handled. Finally, Section 10 and Section 11 respectively discuss privacy and security considerations when outsourcing the Homenet Zone.

### 3. Terminology

- Customer Premises Equipment: (CPE) is a router providing connectivity to the home network.
- Homenet Naming Authority: (HNA) is a home network node responsible to manage the Homenet Zone. This includes building the Homenet Zone, as well as managing the distribution of that Homenet Zone through the Outsourcing Infrastructure.
- Registered Homenet Domain: is the Domain Name associated to the home network.
- Homenet Zone: is the DNS zone associated with the home network. It is designated by its Registered Homenet Domain. This zone



is built by the HNA and contains the bindings between names and IP addresses of the nodes in the home network. The HNA synchronizes the Homenet Zone with the Synchronization Server via a hidden primary / secondary architecture. The Outsourcing Infrastructure may process the Homenet Zone - for example providing DNSSEC signing - to generate the Public Homenet Zone. This Public Homenet Zone is then transmitted to the Public Authoritative Server(s) that publish it on the Internet.

- Public Homenet Zone:    is the public version of the Homenet Zone. It is expected to be signed with DNSSEC. It is hosted by the Public Authoritative Server(s), which are authoritative for this zone. The Public Homenet Zone and the Homenet Zone might be different. For example some names might not become reachable from the Internet, and thus not be hosted in the Public Homenet Zone. Another example of difference may also occur when the Public Homenet Zone is signed whereas the Homenet Zone is not signed.
- Outsourcing Infrastructure:    is the combination of the Synchronization Server and the Public Authoritative Server(s).
- Public Authoritative Servers:    are the authoritative name servers hosting the Public Homenet Zone. Name resolution requests for the Homenet Domain are sent to these servers. For resiliency the Public Homenet Zone SHOULD be hosted on multiple servers.
- Synchronization Server:    is the server with which the HNA synchronizes the Homenet Zone. The Synchronization Server is configured as a secondary and the HNA acts as primary. There MAY be multiple Synchronization Servers, but the text assumes a single server. In addition, the text assumes the Synchronization Server is a separate entity. This is not a requirement, and when the HNA signs the zone, the synchronization function might also be operated by the Public Authoritative Servers.
- Homenet Reverse Zone:    The reverse zone file associated with the Homenet Zone.
- Reverse Public Authoritative Servers:    are the authoritative name server(s) hosting the Public Homenet Reverse Zone. Queries for reverse resolution of the Homenet Domain are sent to this server. Similarly to Public Authoritative Servers, for resiliency, the Homenet Reverse Zone SHOULD be hosted on multiple servers.

- Reverse Synchronization Server:    is the server with which the HNA synchronizes the Homenet Reverse Zone. It is configured as a secondary and the HNA acts as primary. There MAY be multiple Reverse Synchronization Servers, but the text assumes a single server. In addition, the text assumes the Reverse Synchronization Server is a separate entity. This is not a requirement, and when the HNA signs the zone, the synchronization function might also be operated by the Reverse Public Authoritative Servers.
- Hidden Primary:    designates the primary server of the HNA, that synchronizes the Homenet Zone with the Synchronization Server. A primary / secondary architecture is used between the HNA and the Synchronization Server. The hidden primary is not expected to serve end user queries for the Homenet Zone as a regular primary server would. The hidden primary is only known to its associated Synchronization Server.

#### 4. Architecture Description

This section describes the architecture for outsourcing the authoritative naming service from the HNA to the Outsourcing Infrastructure. Section 4.1 describes the architecture, Section 4.2 and Section 4.3 illustrates this architecture and shows how the Homenet Zone should be built by the HNA. It also lists the necessary parameters the HNA needs to be able to outsource the authoritative naming service. These two sections are informational and non-normative.

##### 4.1. Architecture Overview

Figure 1 provides an overview of the architecture.

The home network is designated by the Registered Homenet Domain Name -- example.com in Figure 1. The HNA builds the Homenet Zone associated with the home network. How the Homenet Zone is built is out of the scope of this document. The HNA may host or interact with multiple services to determine name-to-address mappings, such as a web GUI, DHCP [RFC6644] or mDNS [RFC6762]. These services may coexist and may be used to populate the Homenet Zone. This document assumes the Homenet Zone has been populated with domain names that are intended to be publicly published and that are publicly reachable. More specifically, names associated with services or devices that are not expected to be reachable from outside the home network or names bound to non-globally reachable IP addresses MUST NOT be part of the Homenet Zone.

Once the Homenet Zone has been built, the HNA does not host an authoritative naming service, but instead outsources it to the Outsourcing Infrastructure. The Outsourcing Infrastructure takes the Homenet Zone as an input and publishes the Public Homenet Zone. If the HNA does not sign the Homenet Zone, the Outsourcing Infrastructure may instead sign it on behalf of the HNA. Figure 1 provides a more detailed description of the Outsourcing Infrastructure, but overall, it is expected that the HNA provides the Homenet Zone. Then the Public Homenet Zone is derived from the Homenet Zone and published on the Internet.

As a result, DNS queries from the DNS resolvers on the Internet are answered by the Outsourcing Infrastructure and do not reach the HNA. Figure 1 illustrates the case of the resolution of `node1.example.com`.

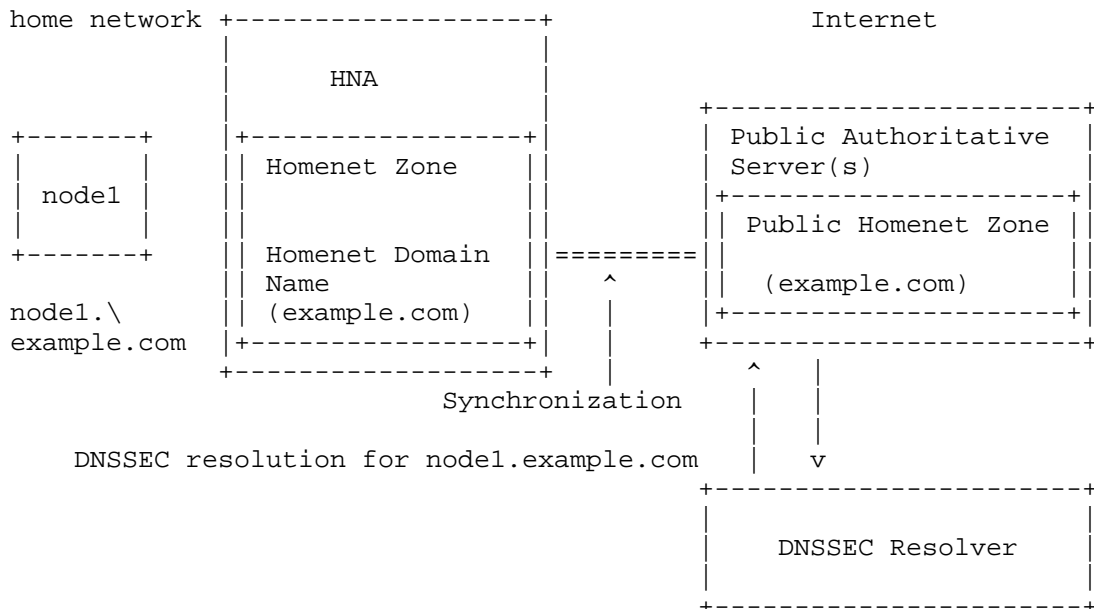


Figure 1: Homenet Naming Architecture Description

The Outsourcing Infrastructure is described in Figure 2. The Synchronization Server receives the Homenet Zone as an input. The received zone may be transformed to output the Public Homenet Zone. Various operations may be performed here, however this document only considers zone signing as a potential operation. This should occur only when the HNA outsources this operation to the Synchronization Server. On the other hand, if the HNA signs the Homenet Zone itself, the zone would be collected by the Synchronization Server and

directly transferred to the Public Authoritative Server(s). These policies are discussed and detailed in Section 6 and Section 7.

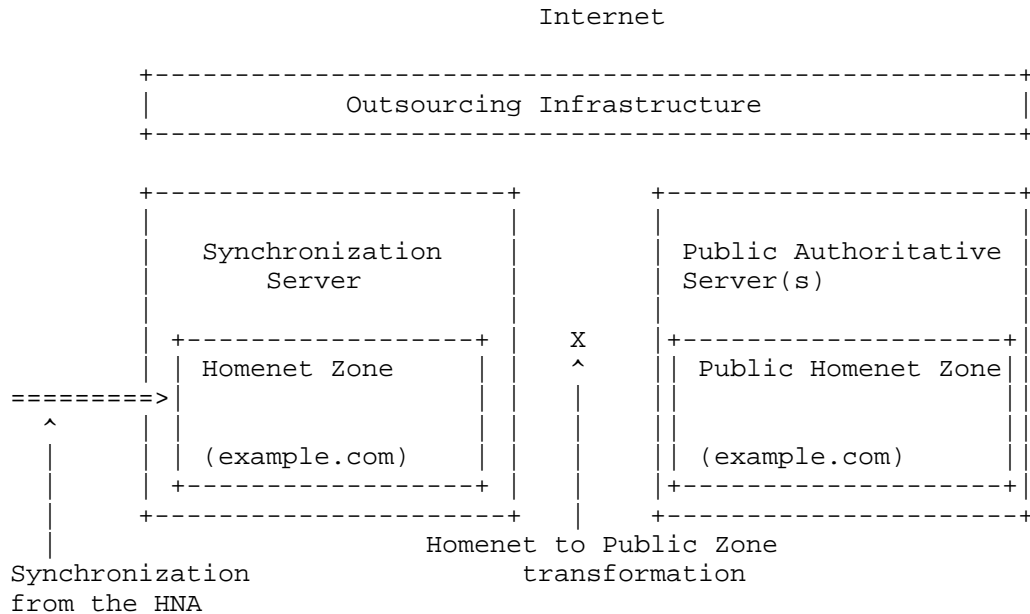


Figure 2: Outsourcing Infrastructure Description

#### 4.2. Example: Homenet Zone

This section is not normative and intends to illustrate how the HNA builds the Homenet Zone.

As depicted in Figure 1 and Figure 2, the Public Homenet Zone is hosted on the Public Authoritative Server(s), whereas the Homenet Zone is hosted on the HNA. Motivations for keeping these two zones identical are detailed in Section 7, and this section considers that the HNA builds the zone that will be effectively published on the Public Authoritative Server(s). In other words "Homenet to Public Zone transformation" is the identity also commonly designated as "no operation" (NOP).

In that case, the Homenet Zone should configure its Name Server RRset (NS) and Start of Authority (SOA) with the values associated with the Public Authoritative Server(s). This is illustrated in Figure 3. `public.primary.example.net` is the FQDN of the Public Authoritative Server(s), and `IP1`, `IP2`, `IP3`, `IP4` are the associated IP addresses. Then the HNA should add the additional new nodes that enter the home

network, remove those that should be removed, and sign the Homenet Zone.

```
$ORIGIN example.com
$TTL 1h

@ IN SOA public.primary.example.net
    hostmaster.example.com. (
        2013120710 ; serial number of this zone file
        1d         ; secondary refresh
        2h         ; secondary retry time in case of a problem
        4w         ; secondary expiration time
        1h         ; maximum caching time in case of failed
                   ; lookups
    )

@ NS public.authoritative.servers.example.net

public.primary.example.net A @IP1
public.primary.example.net A @IP2
public.primary.example.net AAAA @IP3
public.primary.example.net AAAA @IP4
```

Figure 3: Homenet Zone

The SOA RRset is defined in [RFC1033], [RFC1035] and [RFC2308]. This SOA is specific, as it is used for the synchronization between the Hidden Primary and the Synchronization Server and published on the DNS Public Authoritative Server(s)..

- MNAME: indicates the primary. In our case the zone is published on the Public Authoritative Server(s), and its name MUST be included. If multiple Public Authoritative Server(s) are involved, one of them MUST be chosen. More specifically, the HNA MUST NOT include the name of the Hidden Primary.
- RNAME: indicates the email address to reach the administrator. [RFC2142] recommends using hostmaster@domain and replacing the '@' sign by '.'.
- REFRESH and RETRY: indicate respectively in seconds how often secondaries need to check the primary, and the time between two refresh when a refresh has failed. Default values indicated by [RFC1033] are 3600 (1 hour) for refresh and 600 (10 minutes) for retry. This value might be too long for highly dynamic content. However, the Public Authoritative Server(s) and the HNA are expected to implement NOTIFY [RFC1996]. So whilst shorter refresh timers might increase the bandwidth usage for

secondaries hosting large number of zones, it will have little practical impact on the elapsed time required to achieve synchronization between the Outsourcing Infrastructure and the Hidden Master. As a result, the default values are acceptable.

**EXPIRE:**    is the upper limit data SHOULD be kept in absence of refresh. The default value indicated by [RFC1033] is 3600000 (approx. 42 days). In home network architectures, the HNA provides both the DNS synchronization and the access to the home network. This device may be plugged and unplugged by the end user without notification, thus we recommend a long expiry timer.

**MINIMUM:**   indicates the minimum TTL. The default value indicated by [RFC1033] is 86400 (1 day). For home network, this value MAY be reduced, and 3600 (1 hour) seems more appropriate.

#### 4.3. Example: HNA necessary parameters for outsourcing

This section specifies the various parameters required by the HNA to configure the naming architecture of this document. This section is informational, and is intended to clarify the information handled by the HNA and the various settings to be done.

Synchronization Server may be configured with the following parameters. These parameters are necessary to establish a secure channel between the HNA and the Synchronization Server as well as to specify the DNS zone that is in the scope of the communication:

- Synchronization Server:    The associated FQDNs or IP addresses of the Synchronization Server. IP addresses are optional and the FQDN is sufficient. To secure the binding name and IP addresses, a DNSSEC exchange is required. Otherwise, the IP addresses should be entered manually.
- Authentication Method:    How the HNA authenticates itself to the Synchronization Server. This MAY depend on the implementation but this should cover at least IPsec, DTLS and TSIG
- Authentication data:    Associated Data. PSK only requires a single argument. If other authentication mechanisms based on certificates are used, then HNA private keys, certificates and certification authority should be specified.
- Public Authoritative Server(s):    The FQDN or IP addresses of the Public Authoritative Server(s). It MAY correspond to the data that will be set in the NS RRsets and SOA of the Homenet Zone. IP addresses are optional and the FQDN is sufficient. To

secure the binding between name and IP addresses, a DNSSEC exchange is required. Otherwise, the IP addresses should be entered manually.

- Registered Homenet Domain:    The domain name used to establish the secure channel. This name is used by the Synchronization Server and the HNA for the primary / secondary configuration as well as to index the NOTIFY queries of the HNA when the HNA has been renumbered.

Setting the Homenet Zone requires the following information.

- Registered Homenet Domain:    The Domain Name of the zone. Multiple Registered Homenet Domains may be provided. This will generate the creation of multiple Public Homenet Zones.
- Public Authoritative Server(s):    The Public Authoritative Server(s) associated with the Registered Homenet Domain. Multiple Public Authoritative Server(s) may be provided.

## 5. Synchronization between HNA and the Synchronization Server

The Homenet Reverse Zone and the Homenet Zone MAY be updated either with DNS UPDATE [RFC2136] or using a primary / secondary synchronization. The primary / secondary mechanism is preferred as it scales better and avoids DoS attacks: First the primary notifies the secondary that the zone must be updated and leaves the secondary to proceed with the update when possible. Then, a NOTIFY message is sent by the primary, which is a small packet that is less likely to load the secondary. Finally, the AXFR query performed by the secondary is a small packet sent over TCP (section 4.2 [RFC5936]), which mitigates reflection attacks using a forged NOTIFY. On the other hand, DNS UPDATE (which can be transported over UDP), requires more processing than a NOTIFY, and does not allow the server to perform asynchronous updates.

This document RECOMMENDS use of a primary / secondary mechanism instead of the use of DNS UPDATE. This section details the primary / secondary mechanism.

### 5.1. Synchronization with a Hidden Primary

Uploading and dynamically updating the zone file on the Synchronization Server can be seen as zone provisioning between the HNA (Hidden Primary) and the Synchronization Server (Secondary Server). This can be handled either in band or out of band.

Note that there is no standard way to distribute a DNS primary between multiple devices. As a result, if multiple devices are candidate for hosting the Hidden Primary, some specific mechanisms should be designed so the home network only selects a single HNA for the Hidden Primary. Selection mechanisms based on HNCP [RFC7788] are good candidates.

The Synchronization Server is configured as a secondary for the Homenet Domain Name. This secondary configuration has been previously agreed between the end user and the provider of the Synchronization Server. In order to set the primary / secondary architecture, the HNA acts as a Hidden Primary Server, which is a regular authoritative DNS Server listening on the WAN interface.

The Hidden Primary Server SHOULD accept SOA [RFC1033], AXFR [RFC1034], and IXFR [RFC1995] queries from its configured secondary DNS server(s). The Hidden Primary Server SHOULD send NOTIFY messages [RFC1996] in order to update Public DNS server zones as updates occur. Because, the Homenet Zones are likely to be small, the HNA MUST implement AXFR and SHOULD implement IXFR.

Hidden Primary Server differs from a regular authoritative server for the home network by:

- Interface Binding:    the Hidden Primary Server listens on the WAN Interface, whereas a regular authoritative server for the home network would listen on the home network interface.
- Limited exchanges:    the purpose of the Hidden Primary Server is to synchronize with the Synchronization Server, not to serve any zones to end users. As a result, exchanges are performed with specific nodes (the Synchronization Server). Further, exchange types are limited. The only legitimate exchanges are: NOTIFY initiated by the Hidden Primary and IXFR or AXFR exchanges initiated by the Synchronization Server. On the other hand, regular authoritative servers would respond to any hosts, and any DNS query would be processed. The HNA SHOULD filter IXFR/AXFR traffic and drop traffic not initiated by the Synchronization Server. The HNA MUST listen for DNS on TCP and UDP and MUST at least allow SOA lookups of the Homenet Zone.

## 5.2. Securing Synchronization

Exchange between the Synchronization Server and the HNA MUST be secured, at least for integrity protection and for authentication.

TSIG [RFC2845] or SIG(0) [RFC2931] MAY be used to secure the DNS communications between the HNA and the Synchronization Server. TSIG



uses a symmetric key which can be managed by TKEY [RFC2930]. Management of the key involved in SIG(0) is performed through zone updates. How keys are rolled over with SIG(0) is out-of-scope of this document. The advantage of these mechanisms is that they are only associated with the DNS application. Not relying on shared libraries eases testing and integration. On the other hand, using TSIG, TKEY or SIG(0) requires these mechanisms to be implemented on the HNA, which adds code and complexity. Another disadvantage is that TKEY does not provide authentication mechanisms.

Protocols like TLS [RFC5246] / DTLS [RFC6347] MAY be used to secure the transactions between the Synchronization Server and the HNA. The advantage of TLS/DTLS is that this technology is widely deployed, and most of the devices already embed TLS/DTLS libraries, possibly also taking advantage of hardware acceleration. Further, TLS/DTLS provides authentication facilities and can use certificates to authenticate the Synchronization Server and the HNA. On the other hand, using TLS/DTLS requires implementing DNS exchanges over TLS/DTLS, as well as a new service port. This document therefore does NOT RECOMMEND this option.

IPsec [RFC4301] IKEv2 [RFC7296] MAY also be used to secure transactions between the HNA and the Synchronization Server. Similarly to TLS/DTLS, most HNAs already embed an IPsec stack, and IKEv2 supports multiple authentication mechanisms via the EAP framework. In addition, IPsec can be used to protect DNS exchanges between the HNA and the Synchronization Server without any modifications of the DNS server or client. DNS integration over IPsec only requires an additional security policy in the Security Policy Database (SPD). One disadvantage of IPsec is that NATs and firewall traversal may be problematic. However, in our case, the HNA is connected to the Internet, and IPsec communication between the HNA and the Synchronization Server should not be impacted by middle boxes.

How the PSK can be used by any of the TSIG, TLS/DTLS or IPsec protocols: Authentication based on certificates implies a mutual authentication and thus requires the HNA to manage a private key, a public key, or certificates, as well as Certificate Authorities. This adds complexity to the configuration especially on the HNA side. For this reason, we RECOMMEND that the HNA MAY use PSK or certificate base authentication, and that the Synchronization Server MUST support PSK and certificate based authentication.

Note also that authentication of message exchanges between the HNA and the Synchronization Server SHOULD NOT use the external IP address of the HNA to index the appropriate keys. As detailed in Section 9, the IP addresses of the Synchronization Server and the Hidden Primary

are subject to change, for example while the network is being renumbered. This means that the necessary keys to authenticate transaction SHOULD NOT be indexed using the IP address, and SHOULD be resilient to IP address changes.

### 5.3. HNA Security Policies

This section details security policies related to the Hidden Primary / Secondary synchronization.

The Hidden Primary, as described in this document SHOULD drop any queries from the home network. This could be implemented via port binding and/or firewall rules. The precise mechanism deployed is out of scope of this document.

The Hidden Primary SHOULD drop any DNS queries arriving on the WAN interface that are not issued from the Synchronization Server.

The Hidden Primary SHOULD drop any outgoing packets other than DNS NOTIFY query, SOA response, IXFR response or AXFR responses.

The Hidden Primary SHOULD drop any incoming packets other than DNS NOTIFY response, SOA query, IXFR query or AXFR query.

The Hidden Primary SHOULD drop any non protected IXFR or AXFR exchange, depending on how the synchronization is secured.

## 6. DNSSEC compliant Homenet Architecture

[RFC7368] in Section 3.7.3 recommends DNSSEC to be deployed on both the authoritative server and the resolver. The resolver side is out of scope of this document, and only the authoritative part of the server is considered.

Deploying DNSSEC requires signing the zone and configuring a secure delegation. As described in Section 4.1, signing can be performed either by the HNA or by the Outsourcing Infrastructure. Section 6.1 details the implications of these two alternatives. Similarly, the secure delegation can be performed by the HNA or by the Outsourcing Infrastructure. Section 6.2 discusses these two alternatives.

### 6.1. Zone Signing

This section discusses the pros and cons when zone signing is performed by the HNA or by the Outsourcing Infrastructure. It is RECOMMENDED that the HNA signs the zone unless there is a strong argument against this, such as a HNA that is not capable of signing

the zone. In that case zone signing MAY be performed by the Outsourcing Infrastructure on behalf of the HNA.

Reasons for signing the zone by the HNA are:

- 1: Keeping the Homenet Zone and the Public Homenet Zone equal to securely optimize DNS resolution. As the Public Zone is signed with DNSSEC, RRsets are authenticated, and thus DNS responses can be validated even though they are not provided by the authoritative server. This provides the HNA the ability to respond on behalf of the Public Authoritative Server(s). This could be useful for example if, in the future, the HNA announces to the home network that the HNA can act as a local authoritative primary or equivalent for the Homenet Zone. Currently the HNA is not expected to receive authoritative DNS queries, as its IP address is not mentioned in the Public Homenet Zone. On the other hand most HNAs host a resolving function, and could be configured to perform a local lookup to the Homenet Zone instead of initiating a DNS exchange with the Public Authoritative Server(s). Note that outsourcing the zone signing operation means that all DNSSEC queries SHOULD be cached to perform a local lookup, otherwise a resolution with the Public Authoritative Server(s) would be performed.
- 2: Keeping the Homenet Zone and the Public Homenet Zone equal to securely address the connectivity disruption independence detailed in [RFC7368] section 4.4.1 and 3.7.5. As local lookups are possible in case of network disruption, communications within the home network can still rely on the DNSSEC service. Note that outsourcing the zone signing operation does not address connectivity disruption independence with DNSSEC. Instead local lookup would provide DNS as opposed to DNSSEC responses provided by the Public Authoritative Server(s).
- 3: Keeping the Homenet Zone and the Public Homenet Zone equal to guarantee coherence between DNS responses. Using a unique zone is one way to guarantee uniqueness of the responses among servers and places. Issues generated by different views are discussed in more details in Section 7.
- 2: Privacy and Integrity of the DNSSEC Homenet Zone are better guaranteed. When the Zone is signed by the HNA, it makes modification of the DNS data -- for example for flow redirection -- impossible. As a result, signing the Homenet Zone by the HNA provides better protection for end user privacy.

Reasons for signing the zone by the Outsourcing Infrastructure are:

- 1: The HNA may not be capable of signing the zone, most likely because its firmware does not support this function. However this reason is expected to become less and less valid over time.
- 2: Outsourcing DNSSEC management operations. Management operations involve key roll-over, which can be performed automatically by the HNA and transparently for the end user. Avoiding DNSSEC management is mostly motivated by bad software implementations.
- 3: Reducing the impact of HNA replacement on the Public Homenet Zone. Unless the HNA private keys can be extracted and stored off-device, HNA hardware replacement will result in an emergency key roll-over. This can be mitigated by using relatively small TTLs.
- 4: Reducing configuration impact on the end user. Unless there are zero configuration mechanisms in place to provide credentials between the new HNA and the Synchronization Server, authentication associations between the HNA and the Synchronization Server would need to be re-configured. As HNA replacement is not expected to happen regularly, end users may not be at ease with such configuration settings. However, mechanisms as described in [I-D.ietf-homenet-naming-architecture-dhc-options] use DHCP Options to outsource the configuration and avoid this issue.
- 5: The Outsourcing Infrastructure is more likely to handle private keys more securely than the HNA. However, having all private keys in one place may also nullify that benefit.

## 6.2. Secure Delegation

Secure delegation is achieved only if the DS RRset is properly set in the parent zone. Secure delegation can be performed by the HNA or the Outsourcing Infrastructures (that is the Synchronization Server or the Public Authoritative Server(s)).

The DS RRset can be updated manually with nsupdate for example. This requires the HNA or the Outsourcing Infrastructure to be authenticated by the DNS server hosting the parent of the Public Homenet Zone. Such a trust channel between the HNA and the parent DNS server may be hard to maintain with HNAs, and thus may be easier to establish with the Outsourcing Infrastructure. In fact, the

Public Authoritative Server(s) may use Automating DNSSEC Delegation Trust Maintenance [RFC7344].

## 7. Handling Different Views

The Homenet Zone provides information about the home network. Some users may be tempted to have provide responses dependent on the origin of the DNS query. More specifically, some users may be tempted to provide a different view for DNS queries originating from the home network and for DNS queries coming from the Internet. Each view could then be associated with a dedicated Homenet Zone. Note that this document does not specify how DNS queries originating from the home network are addressed to the Homenet Zone. This could be done via hosting the DNS resolver on the HNA for example.

This section is not normative. Section 7.1 details why some nodes may only be reachable from the home network and not from the global Internet. Section 7.2 briefly describes the consequences of having distinct views such as a "home network view" and an "Internet view". Finally, Section 7.3 provides guidance on how to resolve names that are only significant in the home network, without creating different views.

### 7.1. Misleading Reasons for Local Scope DNS Zone

The motivation for supporting different views is to provide different answers dependent on the origin of the DNS query, for reasons such as:

- 1: An end user may want to have services not published on the Internet. Services like the HNA administration interface that provides the GUI to administer your HNA might not seem advisable to publish on the Internet. Similarly, services like the mapper that registers the devices of your home network may also not be desirable to be published on the Internet. In both cases, these services should only be known or used by the network administrator. To restrict the access of such services, the home network administrator may choose to publish these pieces of information only within the home network, where it might be assumed that the users are more trusted than on the Internet. Even though this assumption may not be valid, at least this may reduce the surface of any attack.
- 2: Services within the home network may be reachable using non global IP addresses. IPv4 and NAT may be one reason. On the other hand IPv6 may favor link-local or site-local IP addresses. These IP addresses are not significant outside the boundaries of the home network. As a result, they MAY be

published in the home network view, and SHOULD NOT be published in the Public Homenet Zone.

## 7.2. Consequences

Enabling different views leads to a non-coherent naming system. Depending on where resolution is performed, some services will not be available. This may be especially inconvenient with devices with multiple interfaces that are attached both to the Internet via a 3G/4G interface and to the home network via a WLAN interface. Devices may also cache the results of name resolution, and these cached entries may no longer be valid if a mobile device moves between a homenet connection and an internet connection e.g. a device temporarily loses wifi signal and switches to 3G.

Regarding local-scope IP addresses, such devices may end up with poor connectivity. Suppose, for example, that DNS resolution is performed via the WLAN interface attached to the HNA, and the response provides local-scope IP addresses, but the communication is initiated on the 3G/4G interface. Communications with local-scope addresses will be unreachable on the Internet, thus aborting the communication. The same situation occurs if a device is flip / flopping between various WLAN networks.

Regarding DNSSEC, if the HNA does not sign the Homenet Zone and outsources the signing process, the two views are different, because one is protected with DNSSEC whereas the other is not. Devices with multiple interfaces will have difficulty securing the naming resolution, as responses originating from the home network may not be signed.

For devices with all its interfaces attached to a single administrative domain, that is to say the home network, or the Internet. Incoherence between DNS responses may still also occur if the device is able to perform DNS resolutions both using the DNS resolving server of the home network, or one of the ISP. DNS resolution performed via the HNA or the ISP resolver may be different than those performed over the Internet.

## 7.3. Guidance and Recommendations

As documented in Section 7.2, it is RECOMMENDED to avoid different views. If network administrators choose to implement multiple views, impacts on devices' resolution SHOULD be evaluated.

As a consequence, the Homenet Zone is expected to be an exact copy of the Public Homenet Zone. As a result, services that are not expected to be published on the Internet SHOULD NOT be part of the Homenet

Zone, local-scope addresses SHOULD NOT be part of the Homenet Zone, and when possible, the HNA SHOULD sign the Homenet Zone.

The Homenet Zone is expected to host public information only. It is not the scope of the DNS service to define local home network boundaries. Instead, local scope information is expected to be provided to the home network using local scope naming services. mDNS [RFC6762] DNS-SD [RFC6763] are two examples of these services. Currently mDNS is limited to a single link network. However, future protocols are expected to leverage this constraint as pointed out in [RFC7558].

## 8. Homenet Reverse Zone

This section is focused on the Homenet Reverse Zone.

Firstly, all considerations for the Homenet Zone apply to the Homenet Reverse Zone. The main difference between the Homenet Reverse Zone and the Homenet Zone is that the parent zone of the Homenet Reverse Zone is most likely managed by the ISP. As the ISP also provides the IP prefix to the HNA, it may be able to authenticate the HNA using mechanisms outside the scope of this document e.g. the physical attachment point to the ISP network. If the Reverse Synchronization Server is managed by the ISP, credentials to authenticate the HNA for the zone synchronization may be set automatically and transparently to the end user. [I-D.ietf-homenet-naming-architecture-dhc-options] describes how automatic configuration may be performed.

With IPv6, the domain space for IP addresses is so large that reverse zone may be confronted with scalability issues. How the reverse zone is generated is out of scope of this document. [I-D.howard-dnsop-ip6rdns] provides guidance on how to address scalability issues.

## 9. Renumbering

This section details how renumbering is handled by the Hidden Primary server or the Synchronization Server. Both types of renumbering are discussed i.e. "make-before-break" and "break-before-make".

In the make-before-break renumbering scenario, the new prefix is advertised, the network is configured to prepare the transition to the new prefix. During a period of time, the two prefixes old and new coexist, before the old prefix is completely removed. In the break-before-make renumbering scenario, the new prefix is advertised making the old prefix obsolete.

Renumbering has been extensively described in [RFC4192] and analyzed in [RFC7010] and the reader is expected to be familiar with them before reading this section.

### 9.1. Hidden Primary

In a renumbering scenario, the Hidden Primary is informed it is being renumbered. In most cases, this occurs because the whole home network is being renumbered. As a result, the Homenet Zone will also be updated. Although the new and old IP addresses may be stored in the Homenet Zone, we recommend that only the newly reachable IP addresses be published.

To avoid reachability disruption, IP connectivity information provided by the DNS SHOULD be coherent with the IP plane. In our case, this means the old IP address SHOULD NOT be provided via the DNS when it is not reachable anymore. Let for example TTL be the TTL associated with a RRset of the Homenet Zone, it may be cached for TTL seconds. Let  $T\_NEW$  be the time the new IP address replaces the old IP address in the Homenet Zone, and  $T\_OLD\_UNREACHABLE$  the time the old IP is not reachable anymore. In the case of the make-before-break, seamless reachability is provided as long as  $T\_OLD\_UNREACHABLE - T\_NEW > 2 * TTL$ . If this is not satisfied, then devices associated with the old IP address in the home network may become unreachable for  $2 * TTL - (T\_OLD\_UNREACHABLE - T\_NEW)$ . In the case of a break-before-make,  $T\_OLD\_UNREACHABLE = T\_NEW$ , and the device may become unreachable up to  $2 * TTL$ .

Once the Homenet Zone file has been updated on the Hidden Primary, the Hidden Primary needs to inform the Outsourcing Infrastructure that the Homenet Zone has been updated and that the IP address to use to retrieve the updated zone has also been updated. Both notifications are performed using regular DNS exchanges. Mechanisms to update an IP address provided by lower layers with protocols like SCTP [RFC4960], MOBIKE [RFC4555] are not considered in this document.

The Hidden Primary SHOULD inform the Synchronization Server that the Homenet Zone has been updated by sending a NOTIFY payload with the new IP address. In addition, this NOTIFY payload SHOULD be authenticated using SIG(0) or TSIG. When the Synchronization Server receives the NOTIFY payload, it MUST authenticate it. Note that the cryptographic key used for the authentication SHOULD be indexed by the Registered Homenet Domain contained in the NOTIFY payload as well as the RRSIG. In other words, the IP address SHOULD NOT be used as an index. If authentication succeeds, the Synchronization Server MUST also notice the IP address has been modified and perform a reachability check before updating its primary configuration. The routability check MAY be performed by sending a SOA request to the



Hidden Primary using the source IP address of the NOTIFY. This exchange is also secured, and if an authenticated response is received from the Hidden Primary with the new IP address, the Synchronization Server SHOULD update its configuration file and retrieve the Homenet Zone using an AXFR or a IXFR exchange.

Note that the primary reason for providing the IP address is that the Hidden Primary is not publicly announced in the DNS. If the Hidden Primary were publicly announced in the DNS, then the IP address update could have been performed using the DNS as described in Section 9.2.

## 9.2. Synchronization Server

Renumbering of the Synchronization Server results in the Synchronization Server changing its IP address. The Synchronization Server is a secondary, so its renumbering does not impact the Homenet Zone. In fact, exchanges to the Synchronization Server are restricted to the Homenet Zone synchronization. In our case, the Hidden Primary MUST be able to send NOTIFY payloads to the Synchronization Server.

If the Synchronization Server is configured in the Hidden Primary configuration file using a FQDN, then the update of the IP address is performed by DNS. More specifically, before sending the NOTIFY, the Hidden Primary performs a DNS resolution to retrieve the IP address of the secondary.

As described in Section 9.1, the Synchronization Server DNS information SHOULD be coherent with the IP plane. Let TTL be the TTL associated with the Synchronization Server FQDN,  $T_{\text{NEW}}$  the time the new IP address replaces the old one and  $T_{\text{OLD\_UNREACHABLE}}$  the time the Synchronization Server is not reachable anymore with its old IP address. Seamless reachability is provided as long as  $T_{\text{OLD\_UNREACHABLE}} - T_{\text{NEW}} > 2 * \text{TTL}$ . If this condition is not met, the Synchronization Server may be unreachable during  $2 * \text{TTL} - (T_{\text{OLD\_UNREACHABLE}} - T_{\text{NEW}})$ . In the case of a break-before-make,  $T_{\text{OLD\_UNREACHABLE}} = T_{\text{NEW}}$ , and it may become unreachable up to  $2 * \text{TTL}$ .

Some DNS infrastructure uses the IP address to designate the secondary, in which case, other mechanisms must be found. The reason for using IP addresses instead of names is generally to reach an internal interface that is not designated by a FQDN, and to avoid potential bootstrap problems. Such scenarios are considered as out of scope in the case of home networks.

## 10. Privacy Considerations

Outsourcing the DNS Authoritative service from the HNA to a third party raises a few privacy related concerns.

The Homenet Zone contains a full description of the services hosted in the network. These services may not be expected to be publicly shared although their names remain accessible through the Internet. Even though DNS makes information public, the DNS does not expect to make the complete list of services public. In fact, making information public still requires the key (or FQDN) of each service to be known by the resolver in order to retrieve information about the services. More specifically, making mywebsite.example.com public in the DNS, is not sufficient to make resolvers aware of the existence web site. However, an attacker may walk the reverse DNS zone, or use other reconnaissance techniques to learn this information as described in [RFC7707].

In order to prevent the complete Homenet Zone being published on the Internet, AXFR queries SHOULD be blocked on the Public Authoritative Server(s). Similarly, to avoid zone-walking NSEC3 [RFC5155] SHOULD be preferred over NSEC [RFC4034].

When the Homenet Zone is outsourced, the end user should be aware that it provides a complete description of the services available on the home network. More specifically, names usually provides a clear indication of the service and possibly even the device type, and as the Homenet Zone contains the IP addresses associated with the service, they also limit the scope of the scan space.

In addition to the Homenet Zone, the third party can also monitor the traffic associated with the Homenet Zone. This traffic may provide an indication of the services an end user accesses, plus how and when they use these services. Although, caching may obfuscate this information inside the home network, it is likely that outside your home network this information will not be cached.

## 11. Security Considerations

The Homenet Naming Architecture described in this document solves exposing the HNA's DNS service as a DoS attack vector.

### 11.1. Names are less secure than IP addresses

This document describes how an end user can make their services and devices from his home network reachable on the Internet by using names rather than IP addresses. This exposes the home network to attackers, since names are expected to include less entropy than IP

addresses. In fact, with IP addresses, the Interface Identifier is 64 bits long leading to up to  $2^{64}$  possibilities for a given subnetwork. This is not to mention that the subnet prefix is also of 64 bits long, thus providing up to  $2^{64}$  possibilities. On the other hand, names used either for the home network domain or for the devices present less entropy (livebox, router, printer, nicolas, jennifer, ...) and thus potentially exposes the devices to dictionary attacks.

#### 11.2. Names are less volatile than IP addresses

IP addresses may be used to locate a device, a host or a service. However, home networks are not expected to be assigned a time invariant prefix by ISPs. As a result, observing IP addresses only provides some ephemeral information about who is accessing the service. On the other hand, names are not expected to be as volatile as IP addresses. As a result, logging names over time may be more valuable than logging IP addresses, especially to profile an end user's characteristics.

PTR provides a way to bind an IP address to a name. In that sense, responding to PTR DNS queries may affect the end user's privacy. For that reason end users may choose not to respond to PTR DNS queries and MAY instead return a NXDOMAIN response.

#### 11.3. DNS Reflection Attacks

An attacker performs a reflection attack when it sends traffic to one or more intermediary nodes (reflectors), that in turn send back response traffic to the victim. Motivations for using an intermediary node might be anonymity of the attacker, as well as amplification of the traffic. Typically, when the intermediary node is a DNSSEC server, the attacker sends a DNSSEC query and the victim is likely to receive a DNSSEC response. This section analyzes how the different components may be involved as a reflector in a reflection attack. Section 11.3.1 considers the Hidden Primary, Section 11.3.2 the Synchronization Server, and Section 11.3.3 the Public Authoritative Server(s).

##### 11.3.1. Reflection Attack involving the Hidden Primary

With the specified architecture, the Hidden Primary is only expected to receive DNS queries of type SOA, AXFR or IXFR. This section analyzes how these DNS queries may be used by an attacker to perform a reflection attack.

DNS queries of type AXFR and IXFR use TCP and as such are less subject to reflection attacks. This makes SOA queries the only

remaining practical vector of attacks for reflection attacks, based on UDP.

SOA queries are not associated with a large amplification factor compared to queries of type "ANY" or to query of non existing FQDNs. This reduces the probability a DNS query of type SOA will be involved in a DDoS attack.

SOA queries are expected to follow a very specific pattern, which makes rate limiting techniques an efficient way to limit such attacks, and associated impact on the naming service of the home network.

Motivations for such a flood might be a reflection attack, but could also be a resource exhaustion attack performed against the Hidden Primary. The Hidden Primary only expects to exchange traffic with the Synchronization Server, that is its associated secondary. Even though secondary servers may be renumbered as mentioned in Section 9, the Hidden Primary is likely to perform a DNSSEC resolution and find out the associated secondary's IP addresses in use. As a result, the Hidden Primary is likely to limit the origin of its incoming traffic based on the origin IP address.

With filtering rules based on IP address, SOA flooding attacks are limited to forged packets with the IP address of the secondary server. In other words, the only victims are the Hidden Primary itself or the secondary. There is a need for the Hidden Primary to limit that flood to limit the impact of the reflection attack on the secondary, and to limit the resource needed to carry on the traffic by the HNA hosting the Hidden Primary. On the other hand, mitigation should be performed appropriately, so as to limit the impact on the legitimate SOA sent by the secondary.

The main reason for the Synchronization Server sending a SOA query is to update the SOA RRset after the TTL expires, to check the serial number upon the receipt of a NOTIFY query from the Hidden Primary, or to re-send the SOA request when the response has not been received. When a flood of SOA queries is received by the Hidden Primary, the Hidden Primary may assume it is involved in an attack.

There are few legitimate time slots when the secondary is expected to send a SOA query. Suppose  $T\_NOTIFY$  is the time a NOTIFY is sent by the Hidden Primary,  $T\_SOA$  the last time the SOA has been queried,  $TTL$  the TTL associated to the SOA, and  $T\_REFRESH$  the refresh time defined in the SOA RRset. The specific time SOA queries are expected can be for example  $T\_NOTIFY$ ,  $T\_SOA + 2/3\ TTL$ ,  $T\_SOA + TTL$ ,  $T\_SOA + T\_REFRESH$ , and. Outside a few minutes following these specific time slots, the probability that the HNA discards a legitimate SOA query

is very low. Within these time slots, the probability the secondary may have its legitimate query rejected is higher. If a legitimate SOA is discarded, the secondary will re-send SOA query every "retry time" second until "expire time" seconds occurs, where "retry time" and "expire time" have been defined in the SOA.

As a result, it is RECOMMENDED to set rate limiting policies to protect HNA resources. If a flood lasts more than the expired time defined by the SOA, it is RECOMMENDED to re-initiate a synchronization between the Hidden Primary and the secondaries.

#### 11.3.2. Reflection Attacks involving the Synchronization Server

The Synchronization Server acts as a secondary coupled with the Hidden Primary. The secondary expects to receive NOTIFY query, SOA responses, AXFR and IXFR responses from the Hidden Primary.

Sending a NOTIFY query to the secondary generates a NOTIFY response as well as initiating an SOA query exchange from the secondary to the Hidden Primary. As mentioned in [RFC1996], this is a known "benign denial of service attack". As a result, the Synchronization Server SHOULD enforce rate limiting on sending SOA queries and NOTIFY responses to the Hidden Primary. Most likely, when the secondary is flooded with valid and signed NOTIFY queries, it is under a replay attack which is discussed in Section 11.5. The key thing here is that the secondary is likely to be designed to be able to process much more traffic than the Hidden Primary hosted on a HNA.

This paragraph details how the secondary may limit the NOTIFY queries. Because the Hidden Primary may be renumbered, the secondary SHOULD NOT perform permanent IP filtering based on IP addresses. In addition, a given secondary may be shared among multiple Hidden Primaries which make filtering rules based on IP harder to set. The time at which a NOTIFY is sent by the Hidden Primary is not predictable. However, a flood of NOTIFY messages may be easily detected, as a NOTIFY originated from a given Homenet Zone is expected to have a very limited number of unique source IP addresses, even when renumbering is occurring. As a result, the secondary, MAY rate limit incoming NOTIFY queries.

On the Hidden Primary side, it is recommended that the Hidden Primary sends a NOTIFY as long as the zone has not been updated by the secondary. Multiple SOA queries may indicate the secondary is under attack.

### 11.3.3. Reflection Attacks involving the Public Authoritative Servers

Reflection attacks involving the Public Authoritative Server(s) are similar to attacks on any Outsourcing Infrastructure. This is not specific to the architecture described in this document, and thus are considered as out of scope.

In fact, one motivation of the architecture described in this document is to expose the Public Authoritative Server(s) to attacks instead of the HNA, as it is believed that the Public Authoritative Server(s) will be better able to defend itself.

### 11.4. Flooding Attack

The purpose of flooding attacks is mostly resource exhaustion, where the resource can be bandwidth, memory, or CPU for example.

One goal of the architecture described in this document is to limit the surface of attack on the HNA. This is done by outsourcing the DNS service to the Public Authoritative Server(s). By doing so, the HNA limits its DNS interactions between the Hidden Primary and the Synchronization Server. This limits the number of entities the HNA interacts with as well as the scope of DNS exchanges - NOTIFY, SOA, AXFR, IXFR.

The use of an authenticated channel with SIG(0) or TSIG between the HNA and the Synchronization Server, enables detection of illegitimate DNS queries, so appropriate action may be taken - like dropping the queries. If signatures are validated, then most likely, the HNA is under a replay attack, as detailed in Section 11.5

In order to limit the resource required for authentication, it is recommended to use TSIG that uses symmetric cryptography over SIG(0) that uses asymmetric cryptography.

### 11.5. Replay Attack

Replay attacks consist of an attacker either resending or delaying a legitimate message that has been sent by an authorized user or process. As the Hidden Primary and the Synchronization Server use an authenticated channel, replay attacks are mostly expected to use forged DNS queries in order to provide valid traffic.

From the perspective of an attacker, using a correctly authenticated DNS query may not be detected as an attack and thus may generate a response. Generating and sending a response consumes more resources than either dropping the query by the defender, or generating the query by the attacker, and thus could be used for resource exhaustion

attacks. In addition, as the authentication is performed at the DNS layer, the source IP address could be impersonated in order to perform a reflection attack.

Section 11.3 details how to mitigate reflection attacks and Section 11.4 details how to mitigate resource exhaustion. Both sections assume a context of DoS with a flood of DNS queries. This section suggests a way to limit the attack surface of replay attacks.

As SIG(0) and TSIG use inception and expiration time, the time frame for replay attack is limited. SIG(0) and TSIG recommends a fudge value of 5 minutes. This value has been set as a compromise between possibly loose time synchronization between devices and the valid lifetime of the message. As a result, better time synchronization policies could reduce the time window of the attack.

## 12. IANA Considerations

This document has no actions for IANA.

## 13. Acknowledgment

The authors wish to thank Philippe Lemordant for its contributions on the early versions of the draft; Ole Troan for pointing out issues with the IPv6 routed home concept and placing the scope of this document in a wider picture; Mark Townsley for encouragement and injecting a healthy debate on the merits of the idea; Ulrik de Bie for providing alternative solutions; Paul Mockapetris, Christian Jacquenet, Francis Dupont and Ludovic Eschard for their remarks on HNA and low power devices; Olafur Gudmundsson for clarifying DNSSEC capabilities of small devices; Simon Kelley for its feedback as dnsmasq implementer; Andrew Sullivan, Mark Andrew, Ted Lemon, Mikael Abrahamson, Michael Richardson and Ray Bellis for their feedback on handling different views as well as clarifying the impact of outsourcing the zone signing operation outside the HNA; Mark Andrew and Peter Koch for clarifying the renumbering.

## 14. References

### 14.1. Normative References

- [RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <<http://www.rfc-editor.org/info/rfc1034>>.
- [RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <<http://www.rfc-editor.org/info/rfc1035>>.

- [RFC1995] Ohta, M., "Incremental Zone Transfer in DNS", RFC 1995, DOI 10.17487/RFC1995, August 1996, <<http://www.rfc-editor.org/info/rfc1995>>.
- [RFC1996] Vixie, P., "A Mechanism for Prompt Notification of Zone Changes (DNS NOTIFY)", RFC 1996, DOI 10.17487/RFC1996, August 1996, <<http://www.rfc-editor.org/info/rfc1996>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC2136] Vixie, P., Ed., Thomson, S., Rekhter, Y., and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)", RFC 2136, DOI 10.17487/RFC2136, April 1997, <<http://www.rfc-editor.org/info/rfc2136>>.
- [RFC2142] Crocker, D., "Mailbox Names for Common Services, Roles and Functions", RFC 2142, DOI 10.17487/RFC2142, May 1997, <<http://www.rfc-editor.org/info/rfc2142>>.
- [RFC2308] Andrews, M., "Negative Caching of DNS Queries (DNS NCACHE)", RFC 2308, DOI 10.17487/RFC2308, March 1998, <<http://www.rfc-editor.org/info/rfc2308>>.
- [RFC2845] Vixie, P., Gudmundsson, O., Eastlake 3rd, D., and B. Wellington, "Secret Key Transaction Authentication for DNS (TSIG)", RFC 2845, DOI 10.17487/RFC2845, May 2000, <<http://www.rfc-editor.org/info/rfc2845>>.
- [RFC2930] Eastlake 3rd, D., "Secret Key Establishment for DNS (TKEY RR)", RFC 2930, DOI 10.17487/RFC2930, September 2000, <<http://www.rfc-editor.org/info/rfc2930>>.
- [RFC2931] Eastlake 3rd, D., "DNS Request and Transaction Signatures ( SIG(0)s )", RFC 2931, DOI 10.17487/RFC2931, September 2000, <<http://www.rfc-editor.org/info/rfc2931>>.
- [RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, DOI 10.17487/RFC4034, March 2005, <<http://www.rfc-editor.org/info/rfc4034>>.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <<http://www.rfc-editor.org/info/rfc4301>>.



- [RFC4555] Eronen, P., "IKEv2 Mobility and Multihoming Protocol (MOBIKE)", RFC 4555, DOI 10.17487/RFC4555, June 2006, <<http://www.rfc-editor.org/info/rfc4555>>.
- [RFC4960] Stewart, R., Ed., "Stream Control Transmission Protocol", RFC 4960, DOI 10.17487/RFC4960, September 2007, <<http://www.rfc-editor.org/info/rfc4960>>.
- [RFC5155] Laurie, B., Sisson, G., Arends, R., and D. Blacka, "DNS Security (DNSSEC) Hashed Authenticated Denial of Existence", RFC 5155, DOI 10.17487/RFC5155, March 2008, <<http://www.rfc-editor.org/info/rfc5155>>.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <<http://www.rfc-editor.org/info/rfc5246>>.
- [RFC5936] Lewis, E. and A. Hoenes, Ed., "DNS Zone Transfer Protocol (AXFR)", RFC 5936, DOI 10.17487/RFC5936, June 2010, <<http://www.rfc-editor.org/info/rfc5936>>.
- [RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <<http://www.rfc-editor.org/info/rfc6347>>.
- [RFC6644] Evans, D., Droms, R., and S. Jiang, "Rebind Capability in DHCPv6 Reconfigure Messages", RFC 6644, DOI 10.17487/RFC6644, July 2012, <<http://www.rfc-editor.org/info/rfc6644>>.
- [RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, DOI 10.17487/RFC6762, February 2013, <<http://www.rfc-editor.org/info/rfc6762>>.
- [RFC6763] Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", RFC 6763, DOI 10.17487/RFC6763, February 2013, <<http://www.rfc-editor.org/info/rfc6763>>.
- [RFC7296] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., and T. Kivinen, "Internet Key Exchange Protocol Version 2 (IKEv2)", STD 79, RFC 7296, DOI 10.17487/RFC7296, October 2014, <<http://www.rfc-editor.org/info/rfc7296>>.
- [RFC7788] Stenberg, M., Barth, S., and P. Pfister, "Home Networking Control Protocol", RFC 7788, DOI 10.17487/RFC7788, April 2016, <<http://www.rfc-editor.org/info/rfc7788>>.

## 14.2. Informational References

- [I-D.howard-dnsop-ip6rdns]  
Howard, L., "Reverse DNS in IPv6 for Internet Service Providers", draft-howard-dnsop-ip6rdns-00 (work in progress), June 2014.
- [I-D.ietf-homenet-naming-architecture-dhc-options]  
Migault, D., Mrugalski, T., Griffiths, C., Weber, R., and W. Cloetens, "DHCPv6 Options for Homenet Naming Architecture", draft-ietf-homenet-naming-architecture-dhc-options-03 (work in progress), October 2015.
- [RFC1033] Lottor, M., "Domain Administrators Operations Guide", RFC 1033, DOI 10.17487/RFC1033, November 1987, <<http://www.rfc-editor.org/info/rfc1033>>.
- [RFC4192] Baker, F., Lear, E., and R. Droms, "Procedures for Renumbering an IPv6 Network without a Flag Day", RFC 4192, DOI 10.17487/RFC4192, September 2005, <<http://www.rfc-editor.org/info/rfc4192>>.
- [RFC7010] Liu, B., Jiang, S., Carpenter, B., Venaas, S., and W. George, "IPv6 Site Renumbering Gap Analysis", RFC 7010, DOI 10.17487/RFC7010, September 2013, <<http://www.rfc-editor.org/info/rfc7010>>.
- [RFC7344] Kumari, W., Gudmundsson, O., and G. Barwood, "Automating DNSSEC Delegation Trust Maintenance", RFC 7344, DOI 10.17487/RFC7344, September 2014, <<http://www.rfc-editor.org/info/rfc7344>>.
- [RFC7368] Chown, T., Ed., Arkko, J., Brandt, A., Troan, O., and J. Weil, "IPv6 Home Networking Architecture Principles", RFC 7368, DOI 10.17487/RFC7368, October 2014, <<http://www.rfc-editor.org/info/rfc7368>>.
- [RFC7558] Lynn, K., Cheshire, S., Blanchet, M., and D. Migault, "Requirements for Scalable DNS-Based Service Discovery (DNS-SD) / Multicast DNS (mDNS) Extensions", RFC 7558, DOI 10.17487/RFC7558, July 2015, <<http://www.rfc-editor.org/info/rfc7558>>.
- [RFC7707] Gont, F. and T. Chown, "Network Reconnaissance in IPv6 Networks", RFC 7707, DOI 10.17487/RFC7707, March 2016, <<http://www.rfc-editor.org/info/rfc7707>>.

## Appendix A.   Document Change Log

[RFC Editor: This section is to be removed before publication]

-08

- 1: Clarification of the meaning of CPE. The architecture does not consider a single CPE. The CPE represents multiple functions.

-07:

- 1: Ray Hunter is added as a co-author.

-06:

- 2: Ray Hunter is added in acknowledgment.
- 3: Adding Renumbering section with comments from Dallas meeting
- 4: Replacing Master / Primary - Slave / Secondary

Security Consideration has been updated with Reflection attacks, flooding attacks, and replay attacks.

-05:

\*Clarifying on handling different views:

- 1: How the CPE may be involved in the resolution and responds without necessarily requesting the Public Authoritative Server(s) (and eventually the Hidden Primary)
- 2: How to handle local scope resolution that is link-local, site-local and NAT IP addresses as well as Private domain names that the administrator does not want to publish outside the home network.

Adding a Privacy Considerations Section

Clarification on pro/cons outsourcing zone-signing

Documenting how to handle reverse zones

Adding reference to RFC 2308

-04:

\*Clarifications on zone signing

- \*Rewording

- \*Adding section on different views

- \*architecture clarifications

- 03:

- \*Simon's comments taken into consideration

- \*Adding SOA, PTR considerations

- \*Removing DNSSEC performance paragraphs on low power devices

- \*Adding SIG(0) as a mechanism for authenticating the servers

- \*Goals clarification: the architecture described in the document 1) does not describe new protocols, and 2) can be adapted to specific cases for advance users.

- 02:

- \*remove interfaces: "Public Authoritative Server Naming Interface" is replaced by "Public Authoritative Server(s)y(ies)". "Public Authoritative Server Management Interface" is replaced by "Synchronization Server".

- 01.3:

- \*remove the authoritative / resolver services of the CPE.  
Implementation dependent

- \*remove interactions with mdns and dhcp. Implementation dependent.

- \*remove considerations on low powered devices

- \*remove position toward homenet arch

- \*remove problem statement section

- 01.2:

- \* add a CPE description to show that the architecture can fit CPEs

- \* specification of the architecture for very low powered devices.

- \* integrate mDNS and DHCP interactions with the Homenet Naming Architecture.

\* Restructuring the draft. 1) We start from the homenet-arch draft to derive a Naming Architecture, then 2) we show why CPE need mechanisms that do not expose them to the Internet, 3) we describe the mechanisms.

\* I remove the terminology and expose it in the figures A and B.

\* remove the Front End Homenet Naming Architecture to Homenet Naming

-01:

\* Added C. Griffiths as co-author.

\* Updated section 5.4 and other sections of draft to update section on Hidden Primary / Slave functions with CPE as Hidden Primary/Homenet Server.

\* For next version, address functions of MDNS within Homenet Lan and publishing details northbound via Hidden Primary.

-00: First version published.

#### Authors' Addresses

Daniel Migault  
Ericsson  
8400 Boulevard Decarie  
Montreal, QC H4P 2N2  
Canada

Phone: +1 (514) 452-2160  
Email: [daniel.migault@ericsson.com](mailto:daniel.migault@ericsson.com)

Ralf Weber  
Nominum  
2000 Seaport Blvd #400  
Redwood City, CA 94063  
US

Email: [ralf.weber@nominum.com](mailto:ralf.weber@nominum.com)  
URI: <http://www.nominum.com>

Ray Hunter  
Globis Consulting BV  
Weegschaalstraat 3  
5632CW Eindhoven  
The Netherlands

Email: [v6ops@globis.net](mailto:v6ops@globis.net)  
URI: <http://www.globis.net>

Chris Griffiths

Email: [cgriffiths@gmail.com](mailto:cgriffiths@gmail.com)

Wouter Cloetens  
SoftAtHome  
vaartdijk 3 701  
3018 Wijgmaal  
Belgium

Email: [wouter.cloetens@softathome.com](mailto:wouter.cloetens@softathome.com)

Homenet Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: April 17, 2016

M. Stenberg  
Independent  
October 15, 2015

Auto-Configuration of a Network of Hybrid Unicast/Multicast DNS-Based  
Service Discovery Proxy Nodes  
draft-ietf-homenet-hybrid-proxy-zeroconf-02

Abstract

This document describes how a proxy functioning between Unicast DNS-Based Service Discovery and Multicast DNS can be automatically configured using an arbitrary network-level state sharing mechanism.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on April 17, 2016.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Requirements language . . . . .	3
3. Hybrid proxy - what to configure . . . . .	3
3.1. Conflict resolution within network . . . . .	4
3.2. Per-link DNS-SD forward zone names . . . . .	4
3.3. Reasonable defaults . . . . .	5
3.3.1. Network-wide unique link name (scheme 1) . . . . .	5
3.3.2. Node name (scheme 2) . . . . .	5
3.3.3. Link name (scheme 2) . . . . .	5
4. TLVs . . . . .	5
4.1. DNS Delegated Zone TLV . . . . .	6
4.2. Domain Name TLV . . . . .	7
4.3. Node Name TLV . . . . .	7
5. Desirable behavior . . . . .	7
5.1. DNS search path in DHCP requests . . . . .	8
5.2. Hybrid proxy . . . . .	8
5.3. Hybrid proxy network zeroconf daemon . . . . .	8
6. Limited zone stitching for host name resolution . . . . .	8
7. Security Considerations . . . . .	9
8. IANA Considerations . . . . .	9
9. References . . . . .	9
9.1. Normative references . . . . .	9
9.2. Informative references . . . . .	10
Appendix A. Example configuration . . . . .	10
A.1. Used topology . . . . .	10
A.2. Zero-configuration steps . . . . .	11
A.3. TLV state . . . . .	11
A.4. DNS zone . . . . .	12
A.5. Interaction with hosts . . . . .	13
Appendix B. Implementation . . . . .	13
Appendix C. Why not just proxy Multicast DNS? . . . . .	13
C.1. General problems . . . . .	14
C.2. Stateless proxying problems . . . . .	14
C.3. Stateful proxying problems . . . . .	15
Appendix D. Acknowledgements . . . . .	15
Appendix E. Changelog [RFC Editor: please remove] . . . . .	15
Author's Address . . . . .	16

## 1. Introduction

Section 3 ("Hybrid Proxy Operation") of [I-D.ietf-dnssd-hybrid] describes how to translate queries from Unicast DNS-Based Service Discovery described in [RFC6763] to Multicast DNS described in [RFC6762], and how to filter the responses and translate them back to unicast DNS.



This document describes what sort of configuration the participating hybrid proxy servers require, as well as how it can be provided using any network-wide state sharing mechanism such as link-state routing protocol or Home Networking Control Protocol [I-D.ietf-homenet-hncp]. The document also describes a naming scheme which does not even need to be same across the whole covered network to work as long as the specified conflict resolution works. The scheme can be used to provision both forward and reverse DNS zones which employ hybrid proxy for heavy lifting.

This document does not go into low level encoding details of the Type-Length-Value (TLV) data that we want synchronized across a network. Instead, we just specify what needs to be available, and assume every node that needs it has it available.

We go through the mandatory specification of the language used in Section 2, then describe what needs to be configured in hybrid proxies and participating DNS servers across the network in Section 3. How the data is exchanged using arbitrary TLVs is described in Section 4. Finally, some overall notes on desired behavior of different software components is mentioned in Section 5.

## 2. Requirements language

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 [RFC2119].

## 3. Hybrid proxy - what to configure

Beyond the low-level translation mechanism between unicast and multicast service discovery, the hybrid proxy draft [I-D.ietf-dnssd-hybrid] describes just that there have to be NS records pointing to hybrid proxy responsible for each link within the covered network.

In zero-configuration case, choosing the links to be covered is also non-trivial choice; we can use the border discovery functionality (if available) to determine internal and external links. Or we can use some other protocol's presence (or lack of it) on a link to determine internal links within the covered network, and some other signs (depending on the deployment) such as DHCPv6 Prefix Delegation (as described in [RFC3633]) to determine external links that should not be covered.

For each covered link we want forward DNS zone delegation to an appropriate node which is connected to a link, and running hybrid proxy. Therefore the links' forward DNS zone names should be unique

across the network. We also want to populate reverse DNS zone similarly for each IPv4 or IPv6 prefix in use.

There should be DNS-SD browse domain list provided for the network's domain which contains each physical link only once, regardless of how many nodes and hybrid proxy implementations are connected to it.

Yet another case to consider is the list of DNS-SD domains that we want hosts to enumerate for browse domain lists. Typically, it contains only the local network's domain, but there may be also other networks we may want to pretend to be local but are in different scope, or controlled by different organization. For example, a home user might see both home domain's services (TBD-TLD), as well as ISP's services under `isp.example.com`.

### 3.1. Conflict resolution within network

Any naming-related choice on node may have conflicts in the network given that we require only distributed loosely synchronized database. We assume only that the underlying protocol used for synchronization has some concept of precedence between nodes originating conflicting information, and in case of conflict, the higher precedence node MUST keep the name they have chosen. The one(s) with lower precedence MUST either try different one (that is not in use at all according to the current link state information), or choose not to publish the name altogether.

If a node needs to pick a different name, any algorithm works, although simple algorithm choice is just like the one described in Multicast DNS[RFC6762]: append -2, -3, and so forth, until there are no conflicts in the network for the given name.

### 3.2. Per-link DNS-SD forward zone names

How to name the links of a whole network in automated fashion? Two different approaches seem obvious:

1. Unique link name based - `(unique-link).(domain)`.
2. Node and link name - `(link).(unique-node).(domain)`.

The first choice is appealing as it can be much more friendly (especially given manual configuration). For example, it could mean just `lan.example.com` and `wlan.example.com` for a simple home network. The second choice, on the other hand, has a nice property of being local choice as long as node name can be made unique.

The type of naming scheme to use can be left as implementation option. And the actual names themselves SHOULD be also overridable, if the end-user wants to customize them in some way.

### 3.3. Reasonable defaults

Note that any manual configuration, which SHOULD be possible, MUST override the defaults provided here or chosen by the creator of the implementation.

#### 3.3.1. Network-wide unique link name (scheme 1)

It is not obvious how to produce network-wide unique link names for the (unique-link).(domain) scheme. One option would be to base it on type of physical network layer, and then hope that the number of the networks won't be significant enough to confuse (e.g. "lan", or "wlan").

The network-wide unique link names should be only used in small networks. Given a larger network, after conflict resolution, identifying which link is 'lan-42.example.com' may be challenging.

#### 3.3.2. Node name (scheme 2)

Our recommendation is to use some short form which indicates the type of node it is, for example, "openwrt.example.com". As the name is visible to users, it should be kept as short as possible. In theory even more exact model could be helpful, for example, "openwrt-buffalo-wzr-600-dhr.example.com". In practice providing some other records indicating exact node information (and access to management UI) is more sensible.

#### 3.3.3. Link name (scheme 2)

Recommendation for (link) portion of (link).(node).(domain) is to use physical network layer type as base, or possibly even just interface name on the node if it's descriptive enough. For example, "eth0.openwrt.example.com" and "wlan0.openwrt.example.com" may be good enough.

## 4. TLVs

To implement this specification fully, support for following three different TLVs is needed. However, only the DNS Delegated Zone TLVs MUST be supported, and the other two SHOULD be supported.

#### 4.1. DNS Delegated Zone TLV

This TLV is effectively a combined NS and A/AAAA record for a zone. It MUST be supported by implementations conforming to this specification. Implementations SHOULD provide forward zone per link (or optimizing a bit, zone per link with Multicast DNS traffic). Implementations MAY provide reverse zone per prefix using this same mechanism. If multiple nodes advertise same reverse zone, it should be assumed that they all have access to the link with that prefix. However, as noted in Section 5.3, mainly only the node with highest precedence on the link should publish this TLV.

##### Contents:

- o Address field is IPv6 address (e.g. 2001:db8::3) or IPv4 address mapped to IPv6 address (e.g. ::FFFF:192.0.2.1) where the authoritative DNS server for Zone can be found. If the address field is all zeros, the Zone is under global DNS hierarchy and can be found using normal recursive name lookup starting at the authoritative root servers (This is mostly relevant with the S bit below).
- o S-bit indicates that this delegated zone consists of a full DNS-SD domain, which should be used as base for DNS-SD domain enumeration (that is, (field).\_dns-sd.\_udp.(zone) exists). Forward zones MAY have this set. Reverse zones MUST NOT have this set. This can be used to provision DNS search path to hosts for non-local services (such as those provided by ISP, or other manually configured service providers).
- o B-bit indicates that this delegated zone should be included in network's DNS-SD browse list of domains at b.\_dns-sd.\_udp.(domain). Local forward zones SHOULD have this set. Reverse zones SHOULD NOT have this set.
- o L-bit indicates that this delegated zone should be included in the network's DNS-SD legacy browse list of domains at lb.\_dns-sd.\_udp.(DOMAIN-NAME). Local forward zones SHOULD have this bit set, reverse zones SHOULD NOT.
- o Zone is the label sequence of the zone, encoded according to section 3.1. ("Name space definitions") of [RFC1035]. Note that name compression is not required here (and would not have any point in any case), as we encode the zones one by one. The zone MUST end with an empty label.

In case of a conflict (same zone being advertised by multiple parties with different address or bits), conflict should be addressed according to Section 3.1.

#### 4.2. Domain Name TLV

This TLV is used to indicate the base (domain) to be used for the network. If multiple nodes advertise different ones, the conflict resolution rules in Section 3.1 should result in only the one with highest precedence advertising one, eventually. In case of such conflict, user SHOULD be notified somehow about this, if possible, using the configuration interface or some other notification mechanism for the nodes. Like the Zone field in Section 4.1, the Domain Name TLV's contents consist of a single DNS label sequence.

This TLV SHOULD be supported if at all possible. It may be derived using some future DHCPv6 option, or be set by manual configuration. Even on nodes without manual configuration options, being able to read the domain name provided by a different node could make the user experience better due to consistent naming of zones across the network.

By default, if no node advertises domain name TLV, hard-coded default (TBD) should be used.

#### 4.3. Node Name TLV

This TLV is used to advertise a node's name. After the conflict resolution procedure described in Section 3.1 finishes, there should be exactly zero to one nodes publishing each node name. The contents of the TLV should be a single DNS label.

This TLV SHOULD be supported if at all possible. If not supported, and another node chooses to use the (link).(node) naming scheme with this node's name, the contents of the network's domain may look misleading (but due to conflict resolution of per-link zones, still functional).

If the node name has been configured manually, and there is a conflict, user SHOULD be notified somehow about this, if possible, using the configuration interface or some other notification mechanism for the nodes.

#### 5. Desirable behavior

### 5.1. DNS search path in DHCP requests

The nodes following this specification SHOULD provide the used (domain) as one item in the search path to it's hosts, so that DNS-SD browsing will work correctly. They also SHOULD include any DNS Delegated Zone TLVs' zones, that have S bit set.

### 5.2. Hybrid proxy

The hybrid proxy implementation SHOULD support both forward zones, and IPv4 and IPv6 reverse zones. It SHOULD also detect whether or not there are any Multicast DNS entities on a link, and make that information available to the network zeroconf daemon (if implemented separately). This can be done by (for example) passively monitoring traffic on all covered links, and doing infrequent service enumerations on links that seem to be up, but without any Multicast DNS traffic (if so desired).

Hybrid proxy nodes MAY also publish it's own name via Multicast DNS (both forward A/AAAA records, as well as reverse PTR records) to facilitate applications that trace network topology.

### 5.3. Hybrid proxy network zeroconf daemon

The daemon should avoid publishing TLVs about links that have no Multicast DNS traffic to keep the DNS-SD browse domain list as concise as possible. It also SHOULD NOT publish delegated zones for links for which zones already exist by another node with higher precedence.

The daemon (or other entity with access to the TLVs) SHOULD generate zone information for DNS implementation that will be used to serve the (domain) zone to hosts. Domain Name TLV described in Section 4.2 should be used as base for the zone, and then all DNS Delegated Zones described in Section 4.1 should be used to produce the rest of the entries in zone (see Appendix A.4 for example interpretation of the TLVs in Appendix A.3).

## 6. Limited zone stitching for host name resolution

Section 4.1 of the hybrid proxy specification [I-D.ietf-dnssd-hybrid] notes that the stitching of multiple .local zones into a single DNS-SD zone is to be defined later. This specification does not even attempt that, but for the purpose of host name resolution, it is possible to use the set of DNS Delegated Zone TLVs with S-bit or B-bit set to also provide host naming for the (domain). It is done by simply rewriting A/AAAA queries for (name).(domain) to every (name).(ddz-subdomain).(domain), and providing response to the host

when the first non-empty one is received, rewritten back to (name).(domain).

While this scheme is not very scalable, as it multiplies the number of queries by the number of links (given no response in cache), it does work in small networks with relatively few sub-domains.

## 7. Security Considerations

There is a trade-off between security and zero-configuration in general; if used network state synchronization protocol is not authenticated (and in zero-configuration case, it most likely is not), it is vulnerable to local spoofing attacks. We assume that this scheme is used either within (lower layer) secured networks, or with not-quite-zero-configuration initial set-up.

If some sort of dynamic inclusion of links to be covered using border discovery or such is used, then effectively service discovery will share fate with border discovery (and also security issues if any).

## 8. IANA Considerations

This document has no actions for IANA.

## 9. References

### 9.1. Normative references

- [I-D.ietf-dnssd-hybrid] Cheshire, S., "Hybrid Unicast/Multicast DNS-Based Service Discovery", draft-ietf-dnssd-hybrid-00 (work in progress), November 2014.
- [RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <<http://www.rfc-editor.org/info/rfc1035>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, DOI 10.17487/RFC6762, February 2013, <<http://www.rfc-editor.org/info/rfc6762>>.

[RFC6763] Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", RFC 6763, DOI 10.17487/RFC6763, February 2013, <<http://www.rfc-editor.org/info/rfc6763>>.

## 9.2. Informative references

[I-D.ietf-homenet-hnmp]  
Stenberg, M., Barth, S., and P. Pfister, "Home Networking Control Protocol", draft-ietf-homenet-hnmp-09 (work in progress), August 2015.

[RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, DOI 10.17487/RFC3633, December 2003, <<http://www.rfc-editor.org/info/rfc3633>>.

[RFC3646] Droms, R., Ed., "DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3646, DOI 10.17487/RFC3646, December 2003, <<http://www.rfc-editor.org/info/rfc3646>>.

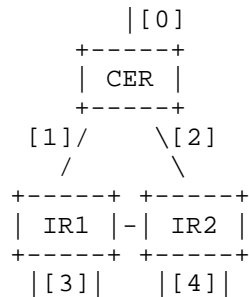
## 9.3. URIs

[1] <https://github.com/sbyx/hnetd/>

## Appendix A. Example configuration

### A.1. Used topology

Let's assume home network that looks like this:



We're not really interested about links [0], [1] and [2], or the links between IRs. Given the optimization described in Section 4.1, they should not produce anything to network's Multicast DNS state (and therefore to DNS either) as there isn't any Multicast DNS traffic there.



The user-visible set of links are [3] and [4]; each consisting of a LAN and WLAN link. We assume that ISP provides 2001:db8:1234::/48 prefix to be delegated in the home via [0].

#### A.2. Zero-configuration steps

Given implementation that chooses to use the second naming scheme (link).(node).(domain), and no configuration whatsoever, here's what happens (the steps are interleaved in practice but illustrated here in order):

1. Network-level state synchronization protocol runs, nodes get effective precedences. For ease of illustration, CER winds up with 2, IR1 with 3, and IR2 with 1.
2. Prefix delegation takes place. IR1 winds up with 2001:db8:1234:11::/64 for LAN and 2001:db8:1234:12::/64 for WLAN. IR2 winds up with 2001:db8:1234:21::/64 for LAN and 2001:db8:1234:22::/64 for WLAN.
3. IR1 is assumed to be reachable at 2001:db8:1234:11::1 and IR2 at 2001:db8:1234:21::1.
4. Each node wants to be called 'node' due to lack of branding in drafts. They announce that using the node name TLV defined in Section 4.3. They also advertise their local zones, but as that information may change, it's omitted here.
5. Conflict resolution ensues. As IR1 has precedence over the rest, it becomes "node". CER and IR2 have to rename, and (depending on timing) one of them becomes "node-2" and other one "node-3". Let us assume IR2 is "node-2". During conflict resolution, each node publishes TLVs for it's own set of delegated zones.
6. CER learns ISP-provided domain "isp.example.com" using DHCPv6 domain list option defined in [RFC3646]. The information is passed along as S-bit enabled delegated zone TLV.

#### A.3. TLV state

Once there is no longer any conflict in the system, we wind up with following TLVs (NN is used as abbreviation for Node Name, and DZ for Delegated Zone TLVs):

```
(from CER)
DZ {s=1,zone="isp.example.com"}

(from IR1)
NN {name="node"}

DZ {address=2001:db8:1234:11::1, b=1,
    zone="lan.node.example.com."}
DZ {address=2001:db8:1234:11::1,
    zone="1.1.0.0.4.3.2.1.8.b.d.0.1.0.0.2.ip6.arpa."}

DZ {address=2001:db8:1234:11::1, b=1,
    zone="wlan.node.example.com."}
DZ {address=2001:db8:1234:11::1,
    zone="2.1.0.0.4.3.2.1.8.b.d.0.1.0.0.2.ip6.arpa."}

(from IR2)
NN {name="node-2"}

DZ {address=2001:db8:1234:21::1, b=1,
    zone="lan.node-2.example.com."}
DZ {address=2001:db8:1234:21::1,
    zone="1.2.0.0.4.3.2.1.8.b.d.0.1.0.0.2.ip6.arpa."}

DZ {address=2001:db8:1234:21::1, b=1,
    zone="wlan.node-2.example.com."}
DZ {address=2001:db8:1234:21::1,
    zone="2.2.0.0.4.3.2.1.8.b.d.0.1.0.0.2.ip6.arpa."}
```

#### A.4. DNS zone

In the end, we should wind up with following zone for (domain) which is example.com in this case, available at all nodes, just based on dumping the delegated zone TLVs as NS+AAAA records, and optionally domain list browse entry for DNS-SD:

```
b._dns_sd._udp PTR lan.node
b._dns_sd._udp PTR wlan.node

b._dns_sd._udp PTR lan.node-2
b._dns_sd._udp PTR wlan.node-2

node AAAA 2001:db8:1234:11::1
node-2 AAAA 2001:db8:1234:21::1

node NS node
node-2 NS node-2
```

```
1.1.0.0.4.3.2.1.8.b.d.0.1.0.0.2.ip6.arpa. NS node.example.com.
2.1.0.0.4.3.2.1.8.b.d.0.1.0.0.2.ip6.arpa. NS node.example.com.
1.2.0.0.4.3.2.1.8.b.d.0.1.0.0.2.ip6.arpa. NS node-2.example.com.
2.2.0.0.4.3.2.1.8.b.d.0.1.0.0.2.ip6.arpa. NS node-2.example.com.
```

Internally, the node may interpret the TLVs as it chooses to, as long as externally defined behavior follows semantics of what's given in the above.

#### A.5. Interaction with hosts

So, what do the hosts receive from the nodes? Using e.g. DHCPv6 DNS options defined in [RFC3646], DNS server address should be one (or multiple) that point at DNS server that has the zone information described in Appendix A.4. Domain list provided to hosts should contain both "example.com" (the hybrid-enabled domain), as well as the externally learned domain "isp.example.com".

When hosts start using DNS-SD, they should check both b.\_dns-sd.\_udp.example.com, as well as b.\_dns-sd.\_udp.isp.example.com for list of concrete domains to browse, and as a result services from two different domains will seem to be available.

#### Appendix B. Implementation

There is an prototype implementation of this draft at [hnetd github repository](#) [1] which contains variety of other homenet WG-related things' implementation too.

#### Appendix C. Why not just proxy Multicast DNS?

Over the time number of people have asked me about how, why, and if we should proxy (originally) link-local Multicast DNS over multiple links.

At some point I meant to write a draft about this, but I think I'm too lazy; so some notes left here for general amusement of people (and to be removed if this ever moves beyond discussion piece).

### C.1. General problems

There are two main reasons why Multicast DNS is not proxyable in the general case.

First reason is the conflict resolution depends on the RRsets staying constant. That is not possible across multiple links (due to e.g. link-local addresses having to be filtered). Therefore, conflict resolution breaks, or at least requires ugly hacks to work around.

A simple, but not really working workaround for this is to make sure that in conflict resolution, propagated resources always loses. Given that the proxy function only removes records, the result SHOULD be consistently original set of records winning. Even with that, the conflict resolution will effectively cease working, allowing for two instances of same name to exist (as both think they 'own' the name due to locally seen higher precedence).

Given some more extra logic, it is possible to make this work by having proxies be aware of both the original record sets, and effectively enforcing the correct conflict resolution results by (for example) passing the unfiltered packets to the losing party just to make sure they renumber, or by altering the RR sets so that they will consistently win (by inserting some lower rrclass/rrtype records). As the conflicts happen only in rrclass=1/rrtype=28, it is easy enough to add e.g. extra TXT record (rrtype 16) to force precedence even when removing the later rrtype 28 record. Obviously, this new RRset must never wind up near the host with the higher precedence, or it will cause spurious renaming loops.

Second reason is timing, which is relatively tight in the conflict resolution phase, especially given lossy and/or high latency networks.

### C.2. Stateless proxying problems

In general, typical stateless proxy has to involve flooding, as Multicast DNS assumes that most messages are received by every host. And it won't scale very well, as a result.

The conflict resolution is also harder without state. It may result in Multicast DNS responder being in constant probe-announce loop, when it receives altered records, notes that it's the one that should own the record. Given stateful proxying, this would be just a

transient problem but designing stateless proxy that won't cause this is non-trivial exercise.

### C.3. Stateful proxying problems

One option is to write proxy that learns state from one link, and propagates it in some way to other links in the network.

A big problem with this case lies in the fact that due to conflict resolution concerns above, it is easy to accidentally send packets that will (possibly due to host mobility) wind up at the originator of the service, who will then perform renaming. That can be alleviated, though, given clever hacks with conflict resolution order.

The stateful proxying may be also too slow to occur within the timeframe allocated for announcing, leading to excessive later renamings based on delayed finding of duplicate services with same name

A work-around exists for this though; if the game doesn't work for you, don't play it. One option would be simply not to propagate ANY records for which conflict has seen even once. This would work, but result in rather fragile, lossy service discovery infrastructure.

There are some other small nits too; for example, Passive Observation Of Failure (POOF) will not work given stateful proxying. Therefore, it leads to requiring somewhat shorter TTLs, perhaps.

### Appendix D. Acknowledgements

Thanks to Stuart Cheshire for the original hybrid proxy draft and interesting discussion in Orlando, where I was finally convinced that stateful Multicast DNS proxying is a bad idea.

Also thanks to Mark Baugher, Ole Troan, Shwetha Bhandari and Gert Doering for review comments.

### Appendix E. Changelog [RFC Editor: please remove]

draft-ietf-homenet-hybrid-proxy-zeroconf-02:

- o Added subsection on simple zone stitching for host naming purposes.

draft-ietf-homenet-hybrid-proxy-zeroconf-01:

- o Refreshed the draft while waiting on progress of draft-ietf-dnssd-hybrid.

Author's Address

Markus Stenberg  
Independent  
Helsinki 00930  
Finland

Email: markus.stenberg@iki.fi

HOMENET  
Internet-Draft  
Intended status: Standards Track  
Expires: February 16, 2017

D. Migault (Ed)  
Ericsson  
T. Mrugalski  
ISC  
C. Griffiths

R. Weber  
Nominum  
W. Cloetens  
SoftAtHome  
August 15, 2016

DHCPv6 Options for Homenet Naming Architecture  
draft-ietf-homenet-naming-architecture-dhc-options-04.txt

Abstract

Home network devices are usually constrained devices with reduced network and CPU capabilities. As such, a home network device exposing the authoritative naming service for its home network on the Internet may become vulnerable to resource exhaustion attacks. One way to avoid exposing these devices is to outsource the authoritative service to a third party, e.g. ISP.

The Homenet Naming Authority (HNA) is the designated device in charge of outsourcing the service to a third party, which requires setting up an architecture.

Such settings may be inappropriate for most end users. This document defines DHCPv6 options so any agnostic HNA can automatically proceed to the appropriate configuration and outsource the authoritative naming service for the home network. In most cases, the outsourcing mechanism is transparent for the end user.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on February 16, 2017.

#### Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

#### Table of Contents

1. Requirements notation . . . . .	3
2. Terminology . . . . .	3
3. Introduction . . . . .	3
4. Protocol Overview . . . . .	6
4.1. Architecture and DHCPv6 Options Overview . . . . .	6
4.2. Mechanisms Securing DNS Transactions . . . . .	9
4.3. Primary / Secondary Synchronization versus DNS Update . .	10
5. HNA Configuration . . . . .	11
5.1. HNA Primary / Secondary Synchronization Configurations .	11
5.1.1. HNA / Synchronization Server . . . . .	11
5.1.2. HNA / Reverse Synchronization Server . . . . .	11
5.2. HNA DNS Data Handling and Update Policies . . . . .	12
5.2.1. Homenet Zone Template . . . . .	12
5.2.2. DNS (Reverse) Homenet Zone . . . . .	12
6. Payload Description . . . . .	13
6.1. Supported Authentication Methods Field . . . . .	13
6.2. Update Field . . . . .	14
6.3. Client Public Key Option . . . . .	14
6.4. Zone Template Option . . . . .	14
6.5. Synchronization Server Option . . . . .	15
6.6. Reverse Synchronization Server Option . . . . .	16
7. DHCP Behavior . . . . .	17
7.1. DHCPv6 Server Behavior . . . . .	17
7.2. DHCPv6 Client Behavior . . . . .	18
7.3. DHCPv6 Relay Agent Behavior . . . . .	18
8. IANA Considerations . . . . .	18



9. Security Considerations . . . . .	19
9.1. DNSSEC is recommended to authenticate DNS hosted data . .	19
9.2. Channel between the HNA and ISP DHCP Server MUST be secured . . . . .	19
9.3. HNAs are sensitive to DoS . . . . .	19
10. Acknowledgments . . . . .	19
11. References . . . . .	19
11.1. Normative References . . . . .	20
11.2. Informational References . . . . .	21
Appendix A. Scenarios and impact on the End User . . . . .	22
A.1. Base Scenario . . . . .	22
A.2. Third Party Registered Homenet Domain . . . . .	23
A.3. Third Party DNS Infrastructure . . . . .	23
A.4. Multiple ISPs . . . . .	25
Appendix B. Document Change Log . . . . .	26
Authors' Addresses . . . . .	27

## 1. Requirements notation

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 [RFC2119].

## 2. Terminology

The reader is expected to be familiar with [I-D.ietf-homenet-front-end-naming-delegation] and its terminology section. This section defines terms that have not been defined in [I-D.ietf-homenet-front-end-naming-delegation]:

- Client Public Key: designates a public key generated by the HNA. This key is used as an authentication credential for the HNA.
- Homenet Zone Template: The template used as a basis to generate the Homenet Zone.
- DNS Template Server: The DNS server that hosts the Homenet Zone Template.
- Homenet Reverse Zone: The reverse zone file associated to the Homenet Zone.

## 3. Introduction

HNAs are usually constrained devices with reduced network and CPU capacities. As such, a HNA hosting on the Internet the authoritative naming service for its home network may become vulnerable to resource exhaustion attacks. Outsourcing the authoritative service to a third

party avoids exposing the HNA to such attacks. This third party can be the ISP or any other independent third party.

Outsourcing the authoritative naming service to a third party requires setting up an architecture designated in this document as the Outsourcing Infrastructure. These settings may be inappropriate for most end users that do not have the sufficient knowledge. To address this issue, this document proposes DHCPv6 options so any agnostic HNA can automatically set the Outsourcing Infrastructure. In most cases, these DHCPv6 options are sufficient and do not require any additional interaction from the end user, thus achieving a zero-config settings. In some other cases, the end user is expected to perform some limited manual configuration.

When the HNA is plugged, the DHCPv6 options described in the document enable:

- 1. To build the Homenet Zone: Building the Homenet Zone requires filling the zone with appropriated bindings such as bindings between the names and the IP addresses of the different devices of the home networks. How the HNA is aware of these binding is out of scope of the document. They may be provided, for example, by the DHCPv6 server hosted on the HNA. On the other hand, building the Homenet Zone also requires configuration parameters like the name of the Registered Domain Name associated to the home network or the Public Authoritative Server(s) the Homenet Zone is outsourced to. These configuration parameters are stored in the Homenet Zone Template. This document describes the Zone Template Option which carries the FQDN associated to the Homenet Zone Template. In order to retrieve the Homenet Zone Template, the HNA sends a query of type AXFR [RFC1034], [RFC5936].
- 2. To upload the Homenet Zone to the Synchronization Server, in charge of publishing the Homenet Zone on the Public Authoritative Server(s). This document describes the Synchronization Server Option that provides the FQDN of the appropriated server. Note that, the document does not consider whether the Homenet Zone is signed or not, and if signed, which entity is responsible to sign it. Such questions are out of the scope of the current document.
- 3. To upload the Homenet Reverse Zone to the Reverse Synchronization Server in charge of publishing the Homenet Reverse Zone on the Reverse Public Authoritative Server(s). This document describes the Reverse Synchronization Server Option that provides the FQDN of the appropriated server. Similarly to item 2., we do not consider in this document if

the Homenet Reverse Zone is signed or not, and if signed who signs it.

- 4. To provide authentication credential (a public key) to the DHCP Server: Information stored in the Homenet Zone Template, the Homenet Zone and Homenet Reverse Zone belongs to the HNA, and only the HNA should be able to update or upload these zones. To authenticate the HNA, this document defines the Client Public Key Option. This option is sent by the HNA to the DHCPv6 server and provides the Client Public Key the HNA uses to authenticate itself. This document does not describe mechanisms used to transmit the Client Public Key from the DHCPv6 server to the appropriate entities. If the DHCPv6 server is not able to provide the Client Public Key to the appropriated entities, then the end user is likely to provide manually the Client Public Key to these entities. This document illustrates two scenarios: one where the DHCPv6 server is responsible for distributing the Client Public Key to the Synchronization Servers and Reverse Synchronization Server. In the other scenarios, the Client Public Key is distributed out of band.

The DHCPv6 options described in this document make possible to configure an Outsourcing Infrastructure with no or little configurations from the end user. A zero-config setting is achieved if the the link between the HNA and the DHCPv6 server and the link between the DHCPv6 server and the various DNS servers (Homenet Zone Server, the Reverse Synchronization Server, Synchronization Server) are trusted. For example, one way to provide a trustworthy connection between the HNA and the DHCPv6 server is defined in [I-D.ietf-dhc-sedhcpv6]. When both links are trusted, the HNA is able to provide its authentication credentials (a Client Public Key) to the DHCPv6 server, that in turn forwards it to the various DNS servers. With the authentication credentials on the DNS servers, the HNA is able to securely update.

If the DHCPv6 server cannot provide the Client Public Key to one of these servers (most likely the Synchronization Server) and the HNA needs to interact with the server, then, the end user is expected to provide the HNA's Client Public Key to these servers (the Reverse Synchronization Server or the Synchronization Server) either manually or using other mechanisms. Such mechanisms are outside the scope of this document. In that case, the authentication credentials need to be provided every time the key is modified. Appendix A provides more details on how different scenarios impact the end users.

The remaining of this document is structured as follows. Section 4 provides an overview of the DHCPv6 options as well as the expected

interactions between the HNA and the various involved entities. This section also provides an overview of available mechanisms to secure DNS transactions and update DNS data. Section 5 describes how the HNA may securely synchronize and update DNS data. Section 6 describes the payload of the DHCPv6 options and Section 7 details how DHCPv6 client, server and relay agent behave. Section 8 lists the new parameters to be registered at the IANA, Section 9 provides security considerations. Finally, Appendix A describes how the HNA may behave and be configured regarding various scenarios.

#### 4. Protocol Overview

This section provides an overview of the HNA's interactions with the Outsourcing Infrastructure in Section 4.1, and so the necessary for its setting. In this document, the configuration is provided via DHCPv6 options. Once configured, the HNA is expected to be able to update and publish DNS data on the different components of the Outsourcing Infrastructure. As a result authenticating and updating mechanisms play an important role in the specification. Section 4.2 provides an overview of the different authentication methods and Section 4.3 provides an overview of the different update mechanisms considered to update the DNS data.

##### 4.1. Architecture and DHCPv6 Options Overview

This section illustrates how a HNA receives the necessary information via DHCPv6 options to outsource its authoritative naming service on the Outsourcing Infrastructure. For the sake of simplicity, this section assumes that the DHCPv6 server is able to communicate to the various DNS servers and to provide them the public key associated with the HNA. Once each server got the public key, the HNA can proceed to transactions in an authenticated and secure way.

This scenario has been chosen as it is believed to be the most popular scenario. This document does not ignore that scenarios where the DHCP Server does not have privileged relations with the Synchronization Server must be considered. These cases are discussed latter in Appendix A. Such scenario does not necessarily require configuration for the end user and can also be zero-config.

The scenario is represented in Figure 1.

- 1: The HNA provides its Client Public Key to the DHCP Server using a Client Public Key Option (OPTION\_PUBLIC\_KEY) and includes the following option codes in its Option Request Option (ORO): Zone Template Option (OPTION\_DNS\_ZONE\_TEMPLATE), the Synchronization Server Option (OPTION\_SYNC\_SERVER) and the

Reverse Synchronization Server Option  
(OPTION\_REVERSE\_SYNC\_SERVER).

- 2: The DHCP Server makes the Client Public Key available to the DNS servers, so the HNA can secure its DNS transactions. How the Client Public Key is transmitted to the various DNS servers is out of scope of this document. Note that the Client Public Key alone is not sufficient to perform the authentication and the key should be, for example, associated with an identifier, or the concerned domain name. How the binding is performed is out of scope of the document. It can be a centralized database or various bindings may be sent to the different servers. Figure 1 represents the specific case where the DHCP Server forwards the set (Client Public Key, Zone Template FQDN) to the DNS Template Server, the set (Client Public Key, IPv6 subnet) to the Reverse Synchronization Server and the set (Client Public Key, Registered Homenet Domain) to the Synchronization Server.
- 3: The DHCP Server responds to the HNA with the requested DHCPv6 options, i.e. the Client Public Key Option (OPTION\_PUBLIC\_KEY), Zone Template Option (OPTION\_DNS\_ZONE\_TEMPLATE), Synchronization Server Option (OPTION\_SYNC\_SERVER), Reverse Synchronization Server Option (OPTION\_REVERSE\_SYNC\_SERVER). Note that this step may be performed in parallel to step 2, or even before. In other words, there is no requirements that step 3 is conducted after step 2.
- 4: Upon receiving the Zone Template Option (OPTION\_DNS\_ZONE\_TEMPLATE), the HNA performs an AXFR DNS query for the Zone Template FQDN. The exchange is authenticated according to the authentication methods defined in the Supported Authentication Methods field of the DHCP option. Once the HNA has retrieved the DNS Zone Template, the HNA can build the Homenet Zone and the Homenet Reverse Zone. Eventually the HNA signs these zones.
- 5: Once the Homenet Reverse Zone has been set, the HNA uploads the zone to the Reverse Synchronization Server. The Reverse Synchronization Server Option (OPTION\_REVERSE\_SYNC\_SERVER) provides the Reverse Synchronization Server FQDN as well as the upload method, and the Supported Authentication Methods protocol to secure the upload.
- 6: Once the Homenet Zone has been set, the HNA uploads the zone to the Synchronization Server. The Synchronization Server Option (OPTION\_SYNC\_SERVER) provides the Synchronization Server FQDN

as well as the upload method and the authentication method to secure the upload.

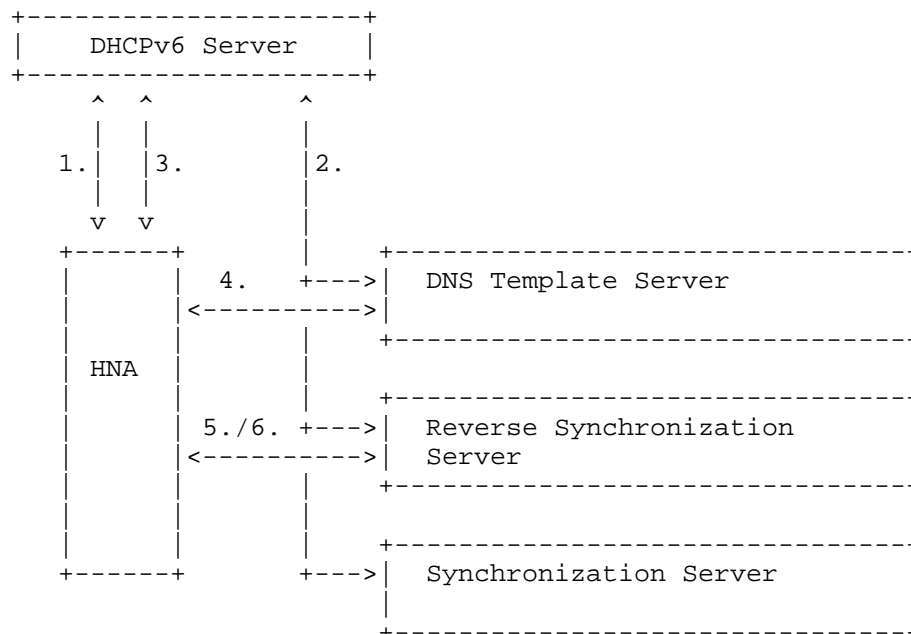


Figure 1: Protocol Overview

As described above, the HNA is likely to interact with various DNS content. More specifically, the HNA is likely to update the:

- Homenet Zone Template: if the configuration of the zone may be changed. This may include additional Public Authoritative Server(s), a different Registered Homenet Domain as the one initially proposed, or a redirection to another domain.
- Homenet Reverse Zone: every time a new device is connected or dis-connected.
- Homenet Zone: every time a new device is connected, dis-connected.

Step 2 and step 3 should be considered as independent steps and could be re-ordered. In fact, the DHCPv6 server does not have to wait for a confirmation from the DNS servers the Client Public Key has been properly received, and is operational by the DNS servers. The DHCP Server is expected to reply upon receiving the Client Public Key Option. The reply to the message with a Client Public Key Option

from the DHCP Server is interpreted by the DHCPv6 client as a confirmation of the reception of the option by the DHCP Server only. It does not indicate whether the server had processed the option or not. Debugging configurations errors or transmission error with one of the DNS servers is let to the HNA and thus is outside of the scope of the DHCPv6. First, it is unlikely a DNS server can validate that the Client Public Key will be operational for the HNA, as multiple causes of errors could occur. For example, the Client Public Key may have been changed during the transmission or by the DHCP Server, or the DNS server may be misconfigured. Second, the number of error codes would be too complex. In addition to multiple causes of errors, multiple architectures and multiple DNS servers may be involved. Third, this may cause significant DHCP Server performance degradation.

In fact, the HNA performs these updates in a secure manner. There are multiple ways to secure a DNS transaction and this document considers two mechanisms: nsupdate and primary/secondary synchronization. Section 4.2 describes the authentication method that may be use to secure the DNS transactions of the HNA. The appropriate authentication methods may, for example, be chosen according to the level of confidentiality or the level of authentication requested by the HNA transactions. Section 4.3 positions the nsupdate and primary/secondary synchronization mechanisms. The update appropriate update mechanism may depend on the for example on the update frequency or the size of the DNS data to update.

#### 4.2. Mechanisms Securing DNS Transactions

Multiple protocols like IPsec [RFC4301] or TLS / DTLS [RFC5246] / [RFC6347] may be used to secure DNS transactions between the HNA and the DNS servers. This document limits its scope to authentication method that have been designed specifically for DNS. This includes DNSSEC [RFC4033], [RFC4034], [RFC4035] that authenticates and provides integrity protection of DNS data, TSIG [RFC2845], [RFC2930] that use a shared secret to secure a transaction between two end points and SIG(0) [RFC2931] authenticates the DNS packet exchanged.

The key issue with TSIG is that a shared secret must be negotiated between the HNA and the server. On the other hand, TSIG performs symmetric cryptography which is light in comparison with asymmetric cryptography used by SIG(0). As a result, over large zone transfer, TSIG may be preferred to SIG(0).

This document does not provide means to distribute shared secret for example using a specific DHCPv6 option. The only assumption made is that the HNA generates or is assigned a public key.

As a result, when the document specifies the transaction is secured with TSIG, it means that either the HNA and the DNS server have been manually configured with a shared secret, or the shared secret has been negotiated using TKEY [RFC2930], and the TKEY exchanged are secured with SIG(0).

Exchanges with the DNS Template Server to retrieve the Homenet Zone Template may be protected by SIG(0), TSIG or DNSSEC. When DNSSEC is used, it means the DNS Template Server only provides integrity protection, and does not necessarily prevent someone else to query the Homenet Zone Template. In addition, DNSSEC is only a way to protect the AXFR queries transaction, in other words, DNSSEC cannot be used to secure updates. If DNSSEC is used to provide integrity protection for the AXFR response, the HNA should proceed to the DNSSEC signature checks. If signature check fails, it MUST reject the response. If the signature check succeeds, the HNA removes all DNSSEC related RRsets (DNSKEY, RRSIG, NSEC\* ...) before building the Homenet Zone. In fact, these DNSSEC related fields are associated to the Homenet Zone Template and not the Homenet Zone.

Any update exchange should use SIG(0) or TSIG to authenticate the exchange.

#### 4.3. Primary / Secondary Synchronization versus DNS Update

As updates only concern DNS zones, this document only considers DNS update mechanisms such as DNS update [RFC2136] [RFC3007] or a primary / secondary synchronization.

The Homenet Zone Template SHOULD be updated with DNS update as it contains static configuration data that is not expected to evolve over time.

The Homenet Reverse Zone and the Homenet Zone can be updated either with DNS update or using a primary / secondary synchronization. As these zones may be large, with frequent updates, we recommend to use the primary / secondary architecture as described in [I-D.ietf-homenet-front-end-naming-delegation]. The primary / secondary mechanism is preferred as it better scales and avoids DoS attacks: First the primary notifies the secondary the zone must be updated, and leaves the secondary to proceed to the update when possible. Then, the NOTIFY message sent by the primary is a small packet that is less likely to load the secondary. At last, the AXFR query performed by the secondary is a small packet sent over TCP (section 4.2 [RFC5936]) which makes unlikely the secondary to perform reflection attacks with a forged NOTIFY. On the other hand, DNS updates can use UDP, packets require more processing than a NOTIFY,



and they do not provide the server the opportunity to postpone the update.

## 5. HNA Configuration

### 5.1. HNA Primary / Secondary Synchronization Configurations

The primary / secondary architecture is described in [I-D.ietf-homenet-front-end-naming-delegation]. The HNA hosts a Hidden Primary that synchronizes with a Synchronization Server or the Reverse Synchronization Server.

When the HNA is plugged its IP address may be unknown to the secondary. The section details how the HNA or primary communicates the necessary information to set up the secondary.

In order to set the primary / secondary configuration, both primary and secondaries must agree on 1) the zone to be synchronized, 2) the IP address and ports used by both primary and secondary.

#### 5.1.1. HNA / Synchronization Server

The HNA is aware of the zone to be synchronized by reading the Registered Homenet Domain in the Homenet Zone Template provided by the Zone Template Option (OPTION\_DNS\_ZONE\_TEMPLATE). The IP address of the secondary is provided by the Synchronization Server Option (OPTION\_SYNC\_SERVER).

The Synchronization Server has been configured with the Registered Homenet Domain and the Client Public Key that identifies the HNA. The only missing information is the IP address of the HNA. This IP address is provided by the HNA by sending a NOTIFY [RFC1996].

When the HNA has built its Homenet Zone, it sends a NOTIFY message to the Synchronization Servers. Upon receiving the NOTIFY message, the secondary reads the Registered Homenet Domain and checks the NOTIFY is sent by the authorized primary. This can be done using the shared secret (TSIG) or the public key (SIG(0)). Once the NOTIFY has been authenticated, the Synchronization Servers might consider the source IP address of the NOTIFY query to configure the primaries attributes.

#### 5.1.2. HNA / Reverse Synchronization Server

The HNA is aware of the zone to be synchronized by looking at its assigned prefix. The IP address of the secondary is provided by the Reverse Synchronization Server Option (OPTION\_REVERSE\_SYNC\_SERVER).

Configuration of the secondary is performed as illustrated in Section 5.1.1.

## 5.2. HNA DNS Data Handling and Update Policies

### 5.2.1. Homenet Zone Template

The Homenet Zone Template contains at least the related fields of the Public Authoritative Server(s) as well as the Homenet Registered Domain, that is SOA, and NS fields. This template might be generated automatically by the owner of the DHCP Server. For example, an ISP might provide a default Homenet Registered Domain as well as default Public Authoritative Server(s). This default settings should provide the HNA the necessary pieces of information to set the homenet naming architecture.

If the Homenet Zone Template is not subject to modifications or updates, the owner of the template might only use DNSSEC to enable integrity check.

On the other hand, the Homenet Zone Template might also be subject to modification by the HNA. The advantage of using the standard DNS zone format is that standard DNS update mechanism can be used to perform updates. These updates might be accepted or rejected by the owner of the Homenet Zone Template. Policies that defines what is accepted or rejected is out of scope of this document. However, this document assumes the Registered Homenet Domain is used as an index by the Synchronization Server, and SIG(0), TSIG are used to authenticate the HNA. As a result, the Registered Homenet Domain should not be modified unless the Synchronization Server can handle with it.

### 5.2.2. DNS (Reverse) Homenet Zone

The Homenet Zone might be generated from the Homenet Zone Template. How the Homenet Zone is generated is out of scope of this document. In some cases, the Homenet Zone might be the exact copy of the Homenet Zone Template. In other cases, it might be generated from the Homenet Zone Template with additional RRsets. In some other cases, the Homenet Zone might be generated without considering the Homenet Zone Template, but only considering specific configuration rules.

In the current document the HNA only sets a single zone that is associated with one single Homenet Registered Domain. The domain might be assigned by the owner of the Homenet Zone Template. This constraint does not prevent the HNA to use multiple domain names. How additional domains are considered is out of scope of this document. One way to handle these additional zones is to configure

static redirections to the Homenet Zone using CNAME [RFC2181], [RFC1034], DNAME [RFC6672] or CNAME+DNAME [I-D.sury-dnsext-cname-dname].

## 6. Payload Description

This section details the payload of the DHCPv6 options. A few DHCPv6 options are used to advertise a server the HNA may be expected to interact with. Interaction may require to define update and authentication methods. Update fields are shared by multiple DHCPv6 options and are described in separate sections. Section 6.1 describes the Supported Authentication Method field, Section 6.2 describes the Update field, the remaining Section 6.3, Section 6.4, Section 6.5, Section 6.6 describe the DHCPv6 options.

### 6.1. Supported Authentication Methods Field

The Supported Authentication Methods field of the DHCPv6 option represented in Figure 2 indicates the authentication method supported by the DNS server. One of these mechanism MUST be chosen by the HNA in order to perform a transaction with the DNS server. See Section 4.2 for more details.

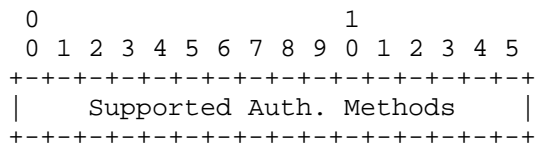


Figure 2: Supported Authentication Methods Filed

- DNS (Bit 0): indicates, when set to 1, that DNS without any security extension is supported.
- DNSSEC (Bit 1): indicates, when set to 1, that DNSSEC provides integrity protection. This can only be used for read operations like retrieving the Homenet Zone Template.
- SIG(0) (Bit 2): indicates, when set to 1, that transaction protected by SIG(0) are supported.
- TSIG (Bit 3): indicates, when set to 1, that transaction using TSIG is supported. Note that if a shared secret has not been previously negotiated between the two party, it should be negotiated using TKEY. The TKEY exchanges MUST be protected with SIG(0) even though SIG(0) is not supported.

- Remaining Bits (Bit 4-15): MUST be set to 0 by the DHCP Server and MUST be ignored by the DHCPv6 client.

A Supported Authentication Methods field with all bits set to zero indicates the operation is not permitted. The Supported Authentication Methods field may be set to zero when updates operations are not permitted for the DNS Homenet Template. In any other case this is an error.

## 6.2. Update Field

The Update Field of the DHCPv6 option is represented in Figure 3. It indicates the update mechanism supported by the DNS server. See Section 4.3 for more details.

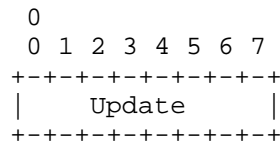


Figure 3: Update Field

- Primary / Secondary (Bit 0): indicates, when set to 1, that DNS Server supports data synchronization using a Primary / Secondary mechanism.
- DNS Update (Bit 1): indicates, when set to 1, that DNS Server supports data synchronization using DNS Updates.
- Remaining Bits (Bit 2-7): MUST be set to 0 by the DHCPv6 server and MUST be ignored by the DHCPv6 client.

## 6.3. Client Public Key Option

The Client Public Key Option (OPTION\_PUBLIC\_KEY) indicates the Client Public Key that is used to authenticate the HNA. This option is defined in [I-D.ietf-dhc-sedhcpv6].

## 6.4. Zone Template Option

The Zone Template Option (OPTION\_DNS\_ZONE\_TEMPLATE) Option indicates the HNA how to retrieve the Homenet Zone Template. It provides a FQDN the HNA SHOULD query with a DNS query of type AXFR as well as the authentication methods associated to the AXFR query or the nsupdate queries. Homenet Zone Template update, if permitted MUST use the DNS Update mechanism.

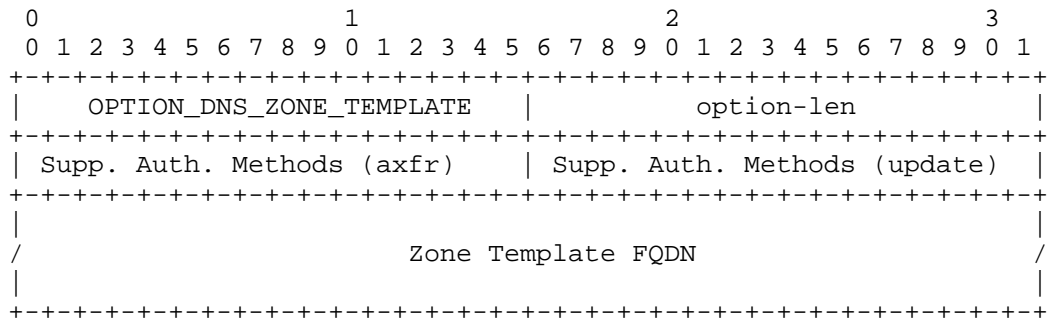


Figure 4: Zone Template Option

- option-code: (16 bits): `OPTION_DNS_ZONE_TEMPLATE`, the option code for the Zone Template Option (TBD1).
- option-len (16 bits): length in octets of the option-data field as described in [RFC3315].
- Supported Authentication Methods(axfr) (16 bits): defines which authentication methods are supported by the DNS server. This field concerns the AXFR and consultation queries, not the update queries. See Section 6.1 for more details.
- Supported Authentication Methods (16 bits): defines which authentication methods are supported by the DNS server. This field concerns the update. See Section 6.1 for more details.
- Zone Template FQDN FQDN (variable): the FQDN of the DNS server hosting the Homenet Zone Template.

#### 6.5. Synchronization Server Option

The Synchronization Server Option (`OPTION_SYNC_SERVER`) provides information necessary for the HNA to upload the Homenet Zone to the Synchronization Server. Finally, the option provides the authentication methods that are available to perform the upload. The upload is performed via a DNS primary / secondary architecture or DNS updates.

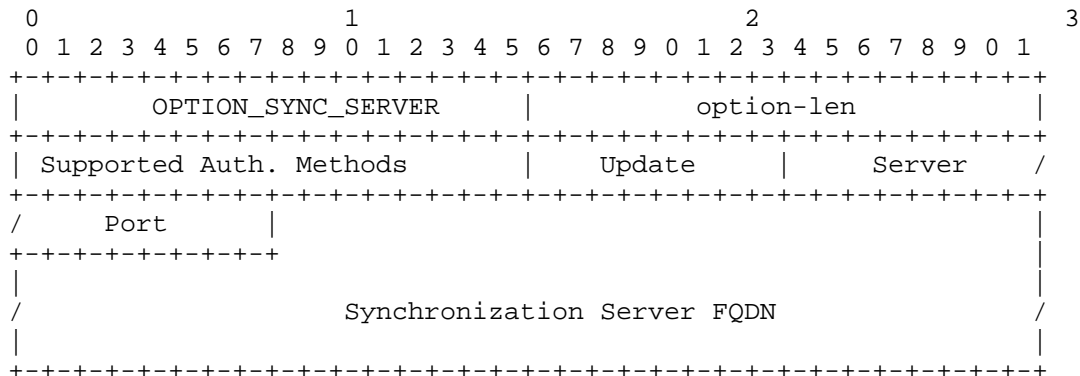


Figure 5: Synchronization Server Option

- option-code (16 bits): `OPTION_SYNC_SERVER`, the option code for the Synchronization Server Option (TBD2).
- option-len (16 bits): length in octets of the option-data field as described in [RFC3315].
- Supported Authentication Methods (16 bits): defines which authentication methods are supported by the DNS server. See Section 6.1 for more details.
- Update (8 bits): defines which update mechanisms are supported by the DNS server. See Section 4.3 for more details.
- Server Port (16 bits): defines the port the Synchronization Server is listening. When multiple transport layers may be used, a single and unique Server Port value applies to all the transport layers. In the case of DNS for example, Server Port value considers DNS exchanges using UDP and TCP.
- Synchronization Server FQDN (variable): the FQDN of the Synchronization Server.

#### 6.6. Reverse Synchronization Server Option

The Reverse Synchronization Server Option (`OPTION_REVERSE_SYNC_SERVER`) provides information necessary for the HNA to upload the Homenet Zone to the Synchronization Server. The option provides the authentication methods that are available to perform the upload. The upload is performed via a DNS primary / secondary architecture or DNS updates.

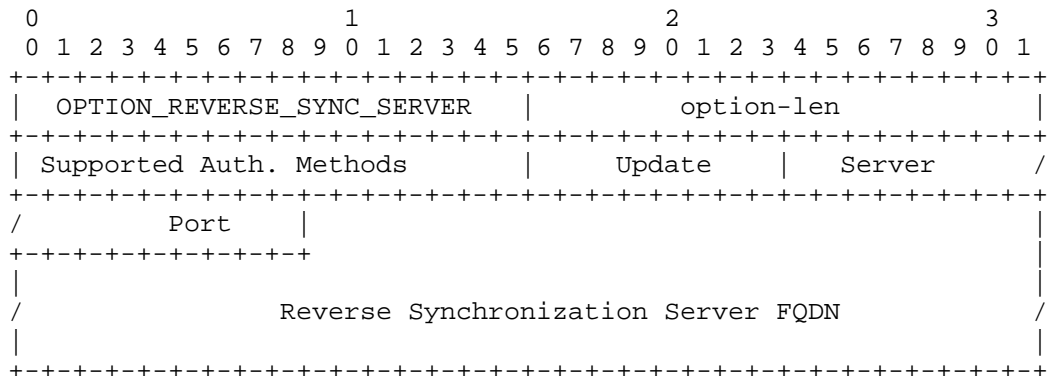


Figure 6: Reverse Synchronization Server Option

- option-code (16 bits): `OPTION_REVERSE_SYNC_SERVER`, the option code for the Reverse Synchronization Server Option (TBD3).
- option-len (16 bits): length in octets of the option-data field as described in [RFC3315].
- Supported Authentication Methods (16 bits): defines which authentication methods are supported by the DNS server. See Section 6.1 for more details.
- Update (8 bits): defines which update mechanisms are supported by the DNS server. See Section 4.3 for more details.
- Server Port (16 bits): defines the port the Synchronization Server is listening.
- Reverse Synchronization Server FQDN (variable): The FQDN of the Reverse Synchronization Server.

## 7. DHCP Behavior

### 7.1. DHCPv6 Server Behavior

Sections 17.2.2 and 18.2 of [RFC3315] govern server operation in regards to option assignment. As a convenience to the reader, we mention here that the server will send option foo only if configured with specific values for foo and if the client requested it. In particular, when configured the DHCP Server sends the Zone Template Option, Synchronization Server Option, Reverse Synchronization Server Option when requested by the DHCPv6 client by including necessary option codes in its ORO.

The DHCP Server may receive a Client Public Key Option (OPTION\_PUBLIC\_KEY) from the HNA. Upon receipt of this DHCPv6 option, the DHCP Server SHOULD acknowledge the reception of the Client Public Key Option as described in Section 4.1 and communicate this credential to the available DNS Servers like the DNS Template Server, the Synchronization Server and the Reverse Synchronization Server, unless not configured to do so.

A HNA may update its Client Public Key by sending a new value in the Client Public Key Option (OPTION\_PUBLIC\_KEY) as this document assumes the link between the HNA and the DHCP Server is considered authenticated and trusted. The server SHOULD process received Client Public Key Option sent by the client (see step 2 in Section 4.1), unless not configured to do so.

## 7.2. DHCPv6 Client Behavior

The DHCPv6 client SHOULD send a Client Public Key Option (OPTION\_PUBLIC\_KEY) to the DHCP Server. This Client Public Key authenticates the HNA.

The DHCPv6 client sends a ORO with the necessary option codes: Zone Template Option, Synchronization Server Option and Reverse Synchronization Server Option.

Upon receiving a DHCP option described in this document in the Reply message, the HNA SHOULD retrieve or update DNS zones using the associated Supported Authentication Methods and update protocols, as described in Section 5.

## 7.3. DHCPv6 Relay Agent Behavior

There are no additional requirements for the DHCP Relay agents.

## 8. IANA Considerations

The DHCP options detailed in this document is:

- OPTION\_DNS\_ZONE\_TEMPLATE: TBD1
- OPTION\_SYNC\_SERVER: TBD2
- OPTION\_REVERSE\_SYNC\_SERVER: TBD3



## 9. Security Considerations

### 9.1. DNSSEC is recommended to authenticate DNS hosted data

It is recommended that the (Reverse) Homenet Zone is signed with DNSSEC. The zone may be signed by the HNA or by a third party. We recommend the zone to be signed by the HNA, and that the signed zone is uploaded.

### 9.2. Channel between the HNA and ISP DHCP Server MUST be secured

The channel MUST be secured because the HNA provides authentication credentials. Unsecured channel may result in HNA impersonation attacks.

The document considers that the channel between the HNA and the ISP DHCP Server is trusted. More specifically, the HNA is authenticated and the exchanged messages are protected. The current document does not specify how to secure the channel. [RFC3315] proposes a DHCP authentication and message exchange protection, [RFC4301], [RFC7296] propose to secure the channel at the IP layer.

### 9.3. HNAs are sensitive to DoS

HNA have not been designed for handling heavy load. The HNA are exposed on the Internet, and their IP address is publicly published on the Internet via the DNS. This makes the Home Network sensitive to Deny of Service Attacks. The resulting outsourcing architecture is described in [I-D.ietf-homenet-front-end-naming-delegation]. This document shows how the outsourcing architecture can be automatically set.

## 10. Acknowledgments

We would like to thank Marcin Siodelski and Bernie Volz for their comments on the design of the DHCPv6 options. We would also like to thank Mark Andrews, Andrew Sullivan and Lorenzo Colliti for their remarks on the architecture design. The designed solution has been largely been inspired by Mark Andrews's document [I-D.andrews-dnsop-pd-reverse] as well as discussions with Mark. We also thank Ray Hunter for its reviews, its comments and for suggesting an appropriated terminology.

## 11. References

### 11.1. Normative References

- [RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <<http://www.rfc-editor.org/info/rfc1034>>.
- [RFC1996] Vixie, P., "A Mechanism for Prompt Notification of Zone Changes (DNS NOTIFY)", RFC 1996, DOI 10.17487/RFC1996, August 1996, <<http://www.rfc-editor.org/info/rfc1996>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC2136] Vixie, P., Ed., Thomson, S., Rekhter, Y., and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)", RFC 2136, DOI 10.17487/RFC2136, April 1997, <<http://www.rfc-editor.org/info/rfc2136>>.
- [RFC2181] Elz, R. and R. Bush, "Clarifications to the DNS Specification", RFC 2181, DOI 10.17487/RFC2181, July 1997, <<http://www.rfc-editor.org/info/rfc2181>>.
- [RFC2845] Vixie, P., Gudmundsson, O., Eastlake 3rd, D., and B. Wellington, "Secret Key Transaction Authentication for DNS (TSIG)", RFC 2845, DOI 10.17487/RFC2845, May 2000, <<http://www.rfc-editor.org/info/rfc2845>>.
- [RFC2930] Eastlake 3rd, D., "Secret Key Establishment for DNS (TKEY RR)", RFC 2930, DOI 10.17487/RFC2930, September 2000, <<http://www.rfc-editor.org/info/rfc2930>>.
- [RFC2931] Eastlake 3rd, D., "DNS Request and Transaction Signatures ( SIG(0)s )", RFC 2931, DOI 10.17487/RFC2931, September 2000, <<http://www.rfc-editor.org/info/rfc2931>>.
- [RFC3007] Wellington, B., "Secure Domain Name System (DNS) Dynamic Update", RFC 3007, DOI 10.17487/RFC3007, November 2000, <<http://www.rfc-editor.org/info/rfc3007>>.
- [RFC3315] Droms, R., Ed., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, DOI 10.17487/RFC3315, July 2003, <<http://www.rfc-editor.org/info/rfc3315>>.

- [RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, DOI 10.17487/RFC4033, March 2005, <<http://www.rfc-editor.org/info/rfc4033>>.
- [RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, DOI 10.17487/RFC4034, March 2005, <<http://www.rfc-editor.org/info/rfc4034>>.
- [RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, DOI 10.17487/RFC4035, March 2005, <<http://www.rfc-editor.org/info/rfc4035>>.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <<http://www.rfc-editor.org/info/rfc4301>>.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <<http://www.rfc-editor.org/info/rfc5246>>.
- [RFC5936] Lewis, E. and A. Hoenes, Ed., "DNS Zone Transfer Protocol (AXFR)", RFC 5936, DOI 10.17487/RFC5936, June 2010, <<http://www.rfc-editor.org/info/rfc5936>>.
- [RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <<http://www.rfc-editor.org/info/rfc6347>>.
- [RFC6672] Rose, S. and W. Wijngaards, "DNAME Redirection in the DNS", RFC 6672, DOI 10.17487/RFC6672, June 2012, <<http://www.rfc-editor.org/info/rfc6672>>.
- [RFC7296] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., and T. Kivinen, "Internet Key Exchange Protocol Version 2 (IKEv2)", STD 79, RFC 7296, DOI 10.17487/RFC7296, October 2014, <<http://www.rfc-editor.org/info/rfc7296>>.

## 11.2. Informational References

- [I-D.andrews-dnsop-pd-reverse] Andrews, M., "Automated Delegation of IP6.ARPA reverse zones with Prefix Delegation", draft-andrews-dnsop-pd-reverse-02 (work in progress), November 2013.

[I-D.ietf-dhc-sedhcpv6]

Jiang, S., Li, L., Cui, Y., Jinmei, T., Lemon, T., and D. Zhang, "Secure DHCPv6", draft-ietf-dhc-sedhcpv6-13 (work in progress), July 2016.

[I-D.ietf-homenet-front-end-naming-delegation]

Migault, D., Weber, R., Hunter, R., Griffiths, C., and W. Cloetens, "Outsourcing Home Network Authoritative Naming Service", draft-ietf-homenet-front-end-naming-delegation-04 (work in progress), September 2015.

[I-D.sury-dnsexst-cname-dname]

Sury, O., "CNAME+DNAME Name Redirection", draft-sury-dnsexst-cname-dname-00 (work in progress), April 2010.

## Appendix A. Scenarios and impact on the End User

This section details various scenarios and discuss their impact on the end user.

### A.1. Base Scenario

The base scenario is the one described in Section 4. It is typically the one of an ISP that manages the DHCP Server, and all DNS servers.

The end user subscribes to the ISP (foo), and at subscription time registers for example.foo as its Registered Homenet Domain example.foo. Since the ISP knows the Registered Homenet Domain and the Public Authoritative Server(s) the ISP is able to build the Homenet Zone Template.

The ISP manages the DNS Template Server, so it is able to load the Homenet Zone Template on the DNS Template Server.

When the HNA is plugged (at least the first time), it provides its Client Public Key to the DHCP Server. In this scenario, the DHCP Server and the DNS Servers are managed by the ISP so the DHCP Server can provide authentication credentials of the HNA to enable secure authenticated transaction between the HNA and these DNS servers. More specifically, credentials are provided to:

- Synchronization Server
- Reverse Synchronization Server
- DNS Template Server

The HNA can update the zone using DNS update or a primary / secondary configuration in a secure way.

The main advantage of this scenario is that the naming architecture is configured automatically and transparently for the end user.

The drawbacks are that the end user uses a Registered Homenet Domain managed by the ISP and that it relies on the ISP naming infrastructure.

#### A.2. Third Party Registered Homenet Domain

This section considers the case when the end user wants its home network to use example.com as a Registered Homenet Domain instead of example.foo that has been assigned by the ISP. We also suppose that example.com is not managed by the ISP.

This can also be achieved without any configuration. When the end user buys the domain name example.com, it may request to redirect the name example.com to example.foo using static redirection with CNAME [RFC2181], [RFC1034], DNAME [RFC6672] or CNAME+DNAME [I-D.sury-dnsex-cname-dname].

This configuration is performed once when the domain name example.com is registered. The only information the end user needs to know is the domain name assigned by the ISP. Once this configuration is done no additional configuration is needed anymore. More specifically, the HNA may be changed, the zone can be updated as in Appendix A.1 without any additional configuration from the end user.

The main advantage of this scenario is that the end user benefits from the Zero Configuration of the Base Scenario Appendix A.1. Then, the end user is able to register for its home network an unlimited number of domain names provided by an unlimited number of different third party providers.

The drawback of this scenario may be that the end user still rely on the ISP naming infrastructure. Note that the only case this may be inconvenient is when the DNS Servers provided by the ISPs results in high latency.

#### A.3. Third Party DNS Infrastructure

This scenario considers that the end user uses example.com as a Registered Homenet Domain, and does not want to rely on the authoritative servers provided by the ISP.

In this section we limit the outsourcing to the Synchronization Server and Public Authoritative Server(s) to a third party. All other DNS Servers DNS Template Server, Reverse Public Authoritative Server(s) and Reverse Synchronization Server remain managed by the ISP. The reason we consider that Reverse Public Authoritative Server(s) and Reverse Synchronization Server remains managed by the ISP are that the prefix is managed by the ISP, so outsourcing these resources requires some redirection agreement with the ISP. More specifically the ISP will need to configure the redirection on one of its Reverse DNS Servers. That said, outsourcing these resources is similar as outsourcing Synchronization Server and Public Authoritative Server(s) to a third party. Similarly, the DNS Template Server can be easily outsourced as detailed in this section

Outsourcing Synchronization Server and Public Authoritative Server(s) requires:

- 1) Updating the Homenet Zone Template: this can be easily done as detailed in Section 4.3 as the DNS Template Server is still managed by the ISP. Such modification can be performed once by any HNA. Once this modification has been performed, the HNA can be changed, the Client Public Key of the HNA may be changed, this does not need to be done another time. One can imagine a GUI on the HNA asking the end user to fill the field with Registered Homenet Domain, optionally Public Authoritative Server(s), with a button "Configure Homenet Zone Template".
- 2) Updating the DHCP Server Information. In fact the Reverse Synchronization Server returned by the ISP is modified. One can imagine a GUI interface that enables the end user to modify its profile parameters. Again, this configuration update is done once-for-ever.
- 3) Upload the authentication credential of the HNA, that is the Client Public Key of the HNA, to the third party. Unless we use specific mechanisms, like communication between the DHCP Server and the third party, or a specific token that is plugged into the HNA, this operation is likely to be performed every time the HNA is changed, and every time the Client Public Key generated by the HNA is changed.

The main advantage of this scenario is that the DNS infrastructure is completely outsourced to the third party. Most likely the Client Public Key that authenticate the HNA need to be configured for every HNA. Configuration is expected to be HNA live-long.

#### A.4. Multiple ISPs

This scenario considers a HNA connected to multiple ISPs.

Firstly, suppose the HNA has been configured with the based scenarios exposed in Appendix A.1. The HNA has multiple interfaces, one for each ISP, and each of these interface is configured using DHCP. The HNA sends to each ISP its Client Public Key Option as well as a request for a Zone Template Option, a Synchronization Server Option and a Reverse Synchronization Server Option. Each ISP provides the requested DHCP options, with different values. Note that this scenario assumes, the home network has a different Registered Homenet Domain for each ISP as it is managed by the ISP. On the other hand, the HNA Client Public Key may be shared between the HNA and the multiple ISPs. The HNA builds the associate DNS(SEC) Homenet Zone, and proceeds to the various settings as described in Appendix A.1.

The protocol and DHCPv6 options described in this document are fully compatible with a HNA connected to multiple ISPs with multiple Registered Homenet Domains. However, the HNA should be able to handle different Registered Homenet Domains. This is an implementation issue which is outside the scope of the current document. More specifically, multiple Registered Homenet Domains leads to multiple DNS(SEC) Homenet Zones. A basic implementation may erase the DNS(SEC) Homenet Zone that exists when it receives DHCPv6 options, and rebuild everything from scratch. This will work for an initial configuration but comes with a few drawbacks. First, updates to the DNS(SEC) Homenet Zone may only push to one of the multiple Registered Homenet Domain, the latest Registered Homenet Domain that has been set, and this is most likely expected to be almost randomly chosen as it may depend on the latency on each ISP network at the boot time. As a results, this leads to unsynchronized Registered Homenet Domains. Secondly, if the HNA handles in some ways resolution, only the latest Registered Homenet Domain set may be able to provide naming resolution in case of network disruption.

Secondly, suppose the HNA is connected to multiple ISP with a single Registered Homenet Domain. In this case, the one party is chosen to host the Registered Homenet Domain. This entity may be one of the ISP or a third party. Note that having multiple ISPs can be motivated for bandwidth aggregation, or connectivity fail-over. In the case of connectivity fail-over, the fail-over concerns the access network and a failure of the access network may not impact the core network where the Synchronization Server and Public Authoritative Primaries are hosted. In that sense, choosing one of the ISP even in a scenario of multiple ISPs may make sense. However, for sake of simplicity, this scenario assumes that a third party has be chosen to host the Registered Homenet Domain. The DNS settings for each ISP is

described in Appendix A.2 and Appendix A.3. With the configuration described in Appendix A.2, the HNA is expect to be able to handle multiple Homenet Registered Domain, as the third party redirect to one of the ISPs Servers. With the configuration described in Appendix A.3, DNS zone are hosted and maintained by the third party. A single DNS(SEC) Homenet Zone is built and maintained by the HNA. This latter configuration is likely to match most HNA implementations.

The protocol and DHCPv6 options described in this document are fully compatible with a HNA connected to multiple ISPs. To configure or not and how to configure the HNA depends on the HNA facilities. Appendix A.1 and Appendix A.2 require the HNA to handle multiple Registered Homenet Domain, whereas Appendix A.3 does not have such requirement.

#### Appendix B. Document Change Log

[RFC Editor: This section is to be removed before publication]

-05: changing Master to Primary, Slave to Secondary

-04: Working Version Major modifications are:

- Re-structuring the draft: description and comparison of update and authentication methods have been integrated into the Overview section. a Configuration section has been created to describe both configuration and corresponding behavior of the HNA.
- Adding Ports parameters: Server Set can configure a port. The Port Server parameter have been added in the DHCPv6 option payloads because middle boxes may not be configured to let port 53 packets and it may also be useful to split servers among different ports, assigning each end user a different port.
- Multiple ISP scenario: In order to address comments, the multiple ISPs scenario has been described to explicitly show that the protocol and DHCPv6 options do not prevent a HNA connected to multiple independent ISPs.

-03: Working Version Major modifications are:

- Redesigning options/scope: according to feed backs received from the IETF89 presentation in the dhc WG.
- Redesigning architecture: according to feed backs received from the IETF89 presentation in the homenet WG, discussion with Mark and Lorenzo.



- 02: Working Version Major modifications are:
  - Redesigning options/scope: As suggested by Bernie Volz
- 01: Working Version Major modifications are:
  - Remove the DNS Zone file construction: As suggested by Bernie Volz
  - DHCPv6 Client behavior: Following options guide lines
  - DHCPv6 Server behavior: Following options guide lines
- 00: version published in the homenet WG. Major modifications are:
  - Reformatting of DHCPv6 options: Following options guide lines
  - DHCPv6 Client behavior: Following options guide lines
  - DHCPv6 Server behavior: Following options guide lines
- 00: First version published in dhc WG.

#### Authors' Addresses

Daniel Migault  
Ericsson  
8400 boulevard Decarie  
Montreal, QC H4P 2N2  
Canada

Email: [daniel.migault@ericsson.com](mailto:daniel.migault@ericsson.com)

Tomek Mrugalski  
Internet Systems Consortium, Inc.  
950 Charter Street  
Redwood City, CA 94063  
USA

Email: [tomasz.mrugalski@gmail.com](mailto:tomasz.mrugalski@gmail.com)  
URI: <http://www.isc.org>

Chris Griffiths

Email: [cgriffiths@gmail.com](mailto:cgriffiths@gmail.com)

Ralf Weber  
Nominum  
2000 Seaport Blvd #400  
Redwood City, CA 94063  
US

Email: [ralf.weber@nominum.com](mailto:ralf.weber@nominum.com)  
URI: <http://www.nominum.com>

Wouter Cloetens  
SoftAtHome  
vaartdijk 3 701  
3018 Wijkmaal  
Belgium

Email: [wouter.cloetens@softathome.com](mailto:wouter.cloetens@softathome.com)

Home Networking  
Internet-Draft  
Updates: RFC7788 (if approved)  
Intended status: Standards Track  
Expires: September 14, 2017

T. Lemon  
Nominum, Inc.  
March 13, 2017

Redacting .home from HNCP  
draft-ietf-homenet-redact-03

## Abstract

This document updates the Home Networking Control Protocol, eliminating the recommendation for a default top-level name for local name resolution.

## Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on September 14, 2017.

## Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Updates to Home Networking Control Protocol . . . . .	2
3. Normative References . . . . .	3
Author's Address . . . . .	3

## 1. Introduction

The Homenet working group has defined a mechanism for sharing information between homenet routers, in Home Networking Control Protocol [2]. That document recommends the use of the ".home" top-level name as a locally-resolved domain name.

RFC7788 did not follow the process defined in Special Use Domain Names [1], or specify how other software should deal with the allocated name. It is likely that, had this process been followed, it would not have been possible to gain consensus on the use of '.home' as the locally-resolved special-use top-level name for homenets, because this name is known to be informally in use by sites on the internet, and the use to which this name has been put is not well documented; it is impossible to say that there are no conflicting uses for the name, and so getting consensus to use it anyway would have been controversial, time consuming, and possibly futile.

The RFC6761 process is not well-understood within the IETF, and the authors of RFC7788 were not aware of it. Normally, authors are not expected to know all there is to know about IETF process, and IETF leadership, specifically working group chairs, area directors and directorate members are expected to engage in a review process that notices oversights of this sort.

Unfortunately, in the case of RFC7788, none of the people who should have caught the missing RFC6761 reference did catch it, and RFC 7788 was published as a consensus document that uses '.home' without ever reserving it in the RFC6761 Special-Use Domain Names registry.

## 2. Updates to Home Networking Control Protocol

This document updates RFC 7788: '.home' MUST NOT be used as the default name for resolution within the home network. The new default value is specified in [3]

### 3. Normative References

- [1] Cheshire, S. and M. Krochmal, "Special-Use Domain Names", RFC 6761, DOI 10.17487/RFC6761, February 2013, <<http://www.rfc-editor.org/info/rfc6761>>.
- [2] Stenberg, M., Barth, S., and P. Pfister, "Home Networking Control Protocol", RFC 7788, DOI 10.17487/RFC7788, April 2016, <<http://www.rfc-editor.org/info/rfc7788>>.
- [3] Pfister, P., "Special Use Top Level Domain '.homenet'", draft-pfister-homenet-dot-00 (work in progress), November 2016.

### Author's Address

Ted Lemon  
Nominum, Inc.  
800 Bridge Parkway  
Redwood City, California 94065  
United States of America

Phone: +1 650 381 6000  
Email: [ted.lemon@nominum.com](mailto:ted.lemon@nominum.com)

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: January 9, 2017

T. Lemon  
Nominum, Inc.  
July 8, 2016

Homenet Naming and Service Discovery Architecture  
draft-lemon-homenet-naming-architecture-01

Abstract

This document recommends a naming and service discovery resolution architecture for homenets. This architecture covers local and global publication of names, discusses security and privacy implications, and addresses those implications. The architecture also covers name resolution and service discovery for hosts on the homenet, and for hosts that roam off of the homenet and still need access to homenet services.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on January 9, 2017.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
1.1. Existing solutions . . . . .	4
2. Terminology . . . . .	5
3. Homenet Naming Database . . . . .	5
3.1. Global Name . . . . .	6
3.2. Local namespaces . . . . .	6
3.3. Public namespaces . . . . .	8
3.4. Maintaining Namespaces . . . . .	9
3.4.1. Multicast DNS . . . . .	9
3.4.2. DNS Update . . . . .	10
3.5. Recovery from loss . . . . .	10
3.6. Well-known names . . . . .	11
4. Name Resolution . . . . .	12
4.1. Configuring Resolvers . . . . .	12
4.2. Configuring Service Discovery . . . . .	12
4.3. Resolution of local namespaces . . . . .	13
4.4. Service Discovery Resolution . . . . .	13
4.5. Local and Public Zones . . . . .	14
4.6. DNSSEC Validation . . . . .	15
4.7. Support for Multiple Provisioning Domains . . . . .	15
4.8. Using the Local Namespace While Away From Home . . . . .	16
5. Publishing the Public Namespace . . . . .	17
5.1. Acquiring the Global Name . . . . .	17
5.2. Hidden Primary/Public Secondaries . . . . .	17
5.3. PKI security . . . . .	18
5.4. Renumbering . . . . .	18
5.5. ULA . . . . .	18
6. Management . . . . .	18
6.1. End-user management . . . . .	18
6.2. Central management . . . . .	18
7. Privacy Considerations . . . . .	19
8. Security Considerations . . . . .	19
9. IANA considerations . . . . .	19
10. Normative References . . . . .	20
Author's Address . . . . .	21

## 1. Introduction

Associating domain names with hosts on the Internet is a key factor in enabling communication with hosts, particularly for service discovery. In order to provide name service, several provisioning mechanisms must be available:

- o Provisioning of a domain name under which names can be published and services advertised
- o Associating names that are subdomains of that name with hosts.
- o Advertising services available on the local network by publishing resource records on those names.
- o Distribution of names published in that namespace to servers that can be queried in order to resolve names
- o Correct advertisement of name servers that can be queried in order to resolve names
- o Timely removal of published names and resource records when they are no longer in use

Homenet adds the following considerations:

1. Some names may be published in a broader scope than others. For example, it may be desirable to advertise some homenet services to users who are not connected to the homenet. However, it is unlikely that all services published on the home network would be appropriate to publish outside of the home network. In many cases, no services will be appropriate to publish outside of the network, but the ability to do so is required.
2. Users cannot be assumed to be skilled or knowledgeable in name service operation, or even to have any sort of mental model of how these functions work. With the possible exception of policy decisions, all of the operations mentioned here must reliably function automatically, without any user intervention or debugging.
3. Even to the extent that users may provide input on policy, such as whether a service should or should not be advertised outside of the home, the user must be able to safely provide such input without having a correct mental model of how naming and service discovery work, and without being able to reason about security in a nuanced way.
4. Because user intervention cannot be required, naming conflicts must be resolved automatically, and, to the extent possible, transparently.
5. Where services are advertised both on and off the home network, differences in naming conventions that may vary depending on the user's location must likewise be transparent to the end user.



6. Hosts that do not implement any homenet-specific capabilities must still be able to discover and access services on the homenet, to the extent possible.
7. Devices that provide services must be able to publish those services on the homenet, and those services must be available from any part of the homenet, not just the link to which the device is attached.
8. Homenet explicitly supports multihoming--connecting to more than one Internet Service Provider--and therefore support for multiple provisioning domains [9] is required to deal with situations where the DNS may give a different answer depending on whether caching resolvers at one ISP or another are queried.
9. Multihomed homenets may treat all service provider links as equivalent, or may treat some links as primary and some as backup, either because of differing transit costs or differing performance. Services advertised off-network may therefore be advertised for some links and not others.
10. To the extent possible, the homenet should support DNSSEC. If the homenet local domain is not unique, there should still be a mechanism that homenet-aware devices can use to bootstrap trust for a particular homenet.

In addition to these considerations, there may be a need to provide for secure communication between end users and the user interface of the home network, as well as to provide secure name validation (e.g., DNSSEC). Secure communications require that the entity being secured have a name that is unique and can be cryptographically authenticated within the scope of use of all devices that must communicate with that entity. Because it is very likely that devices connecting to one homenet will be sufficiently portable that they may connect to many homenets, the scope of use must be assumed to be global. Therefore, each homenet must have a globally unique identifier.

#### 1.1. Existing solutions

Previous attempts to automate naming and service discovery in the context of a home network are able to function with varying degrees of success depending on the topology of the home network. For example, Multicast DNS [7] can provide naming and service discovery [8], but only within a single multicast domain.

The Domain Name System provides a hierarchical namespace [1], a mechanism for querying name servers to resolve names [2], a mechanism for updating namespaces by adding and removing names [4], and a

mechanism for discovering services [8]. Unfortunately, DNS provides no mechanism for automatically provisioning new namespaces, and secure updates to namespaces require pre-shared keys, which won't work for an unmanaged network. DHCP can be used to populate names in a DNS namespace; however at present DHCP cannot provision service discovery information.

Hybrid Multicast DNS [10] proposes a mechanism for extending multicast DNS beyond a single multicast domain.. However, it has serious shortcomings as a solution to the Homenet naming problem. The most obvious shortcoming is that it requires that every multicast domain have a separate name. This then requires that the homenet generate names for every multicast domain, and requires that the end user have a mental model of the topology of the network in order to guess on which link a given service may appear. [xxx is this really true at the UI?]

## 2. Terminology

This document uses the following terms and abbreviations:

HNR Homenet Router

ISP Internet Service Provider

GNRP Global Name Registration Provider

## 3. Homenet Naming Database

In order to resolve names, there must be a place where names are stored. There are two ways to go about this: either names are stored on the devices that own them, or they are stored in the network infrastructure. This isn't a clean division of responsibility, however. It's possible for the device to maintain change control over its own name, while still performing name resolution for that name in the network infrastructure.

If devices maintain change control on their own names, conflicts can arise. Two devices might present the same name, either because their default names or the same, or as a result of accidental. Devices can be attached to more than one link, in which case we want the same name to identify them on both networks. Although homenets are self-configuring, user customization is permitted and useful, and while some devices may provide a user interface for setting their name, it may be worthwhile to provide a user interface and underlying support for allowing the user to specify a device's name in the homenet infrastructure.

In order to achieve this, the Homenet Naming Database (HNDB) provides a persistent central store into which names can be registered.

### 3.1. Global Name

Every homenet must be able to have a name in the global DNS hierarchy which serves as the root of the zone in which the homenet publishes its public namespaces. Homenets that do not yet have a name in the global namespace use the homenet special-use top-level name [TBD1] as their "global name" until they are configured with a global name.

A homenet's global name can be a name that the homenet user has registered on their own in the DNS using a public DNS registrar. However, this is not required and, indeed, presents some operational challenges. It can also be a subdomain of a domain owned by one of the user's ISP, or managed by some DNS service provider that specifically provides homenet naming services.

For most end-users, the second or third options will be preferable. It will allow them to choose an easily-remembered homenet domain name under an easily-remembered service provider subdomain, and will not require them to maintain a DNS registration.

Homenets must support automatic configuration of the homenet global name in a secure manner, as well as manual configuration of the name. The solution must allow a user with a smartphone application or a user with a web browser to successfully configure the homenet's global name without manual data entry. The security implications of this process must be identified and, to the extent possible, addressed.

### 3.2. Local namespaces

Every homenet has two or more non-hierarchical local namespaces, one for names of hosts--the host namespace--and one or more for IP addresses--the address namespaces. A namespace is a database table mapping each of a set keys to its value. "Local" in this context means "visible to users of the homenet," as opposed to "public," meaning visible to anyone.

For the host namespace, the key is the set of labels in a name, excluding whatever labels represent the domain name of the namespace. So for example if the homenet's global name is "dog-pixel.example.com" and the name being looked up is "alice.dog-pixel.example.com", the key will be "alice".

The local namespace may be available both in the global DNS namespace and under the [TBD1] special-use name. The set of keys is the same

in both cases--in the above example, the name could be either 'alice.dog-pixel.example.com' or 'alice.[TBD1]'. Whichever one of the two representations is used, the key is simply 'alice'.

For each address namespace, the key is the locally-significant portion of the IP address. For example, if the local prefix assigned by an ISP is 2001:DB8::/48, the name of that address namespace will be '7.e.e.b.8.b.d.0.1.0.0.2.ip6.arpa'. An IP address of 2001:db8::1 would therefore yield a key of '1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0'

Every prefix in use on the homenet has an address namespace, whether its subdomain is delegated in the DNS or not. This includes any public or private IPv4 prefixes in use [3] as well as any ULA prefixes in use [5], which can't be delegated [6]. When the valid lifetime for a prefix that had been in use on the homenet ends, the address namespace for that prefix is discarded. Namespaces for prefixes that are manually configured, like IPv4 public prefixes and IPv4 private prefixes, persist as long as the prefix is configured. Since ULA prefixes have lifetimes, the lifetime rule applies to their address namespaces.

In all namespaces, the value that the key addresses is a sub-table containing one or more RRsets, each of which is identified by its RRtype. In the terminology of the DNS protocol, each of these namespaces is analogous to a DNS zone (but bear in mind that from the perspective of DNS queries, the namespace for names may appear to hosts connected to the homenet as two different zones containing identical data).

However, in addition to DNS zone data, each RRset also has two metadata flags: the public flag and the critical flag. The public flag indicates whether the data in this RRset should be publicly visible. The critical flag indicates whether the service should be advertised even on high-cost internet links.

Each RR that contains a name (e.g., a CNAME or SRV record) either contains a local name or a name in the public DNS. Local names can be subdomains of the homenet's global name, yet not be public, if no RRsets in the namespace for names is marked public. Local names can also be subdomains of [TBD1]. Names in the public DNS that are not subdomains of the homenet's global name can only be added by explicit action in one of the management interfaces described in Section 6.

Each local namespace is maintained as a distributed database with copies on every homenet router. No copy is the master copy. Although the local namespace is non-hierarchical, it is permissible for it to contain RRtypes that contain delegations. However, from an

operational perspective is is most likely better for the local namespace to be at the bottom of the delegation hierarchy, and so we do not recommend the use of such delegations.

### 3.3. Public namespaces

Every homenet has one or more public namespaces. These are subsets of the local namespaces with the following modifications:

1. Names with no RRsets whose public bits are set are not included in the public namespace.
2. RRs that contain IP addresses in the homenet's ULA prefix are omitted.
3. By default, RRs that contain IPv4 addresses are omitted, because IPv4 doesn't support renumbering. However, there should be a whitelist of IPv4 addresses that may be published, so that if the end user has static IPv4 addresses, those can be published. Private IPv4 addresses, however, are never published.
4. If an RRset is marked best-effort rather than critical, RRs containing IP addresses that have prefixes assigned by backup links are omitted.
5. If an RRset contains names, names that are subdomains of either the homenet's global name or [TBD1] are checked in the local host namespace to see if they are marked public. If not, they are omitted.

Because the public namespaces are subsets of the local namespaces, replication is not necessary: each homenet router automatically produces public namespaces by deriving them from the local namespaces using the above rules. Answers to queries in the public namespaces can be generated on demand. However, it may be preferable to maintain these namespaces as if they were DNS zones. This makes it possible to use DNS zone transfers to offload the contents of public zones to a secondary service provider, eliminating the need to handle arbitrary numbers of queries from off of the homenet.

A mechanism will be present that allows devices that have been configured to publicly advertise services to indicate to the homenet that the public bit and/or the backup bit will be set in RRsets that they publish.

### 3.4. Maintaining Namespaces

Homenets support three methods for maintaining local namespaces. These rely on Multicast DNS, DNS updates, and any of the management mechanisms mentioned in Section 6.

#### 3.4.1. Multicast DNS

HNRs cooperate to maintain a DNS mirror of the set of names published by mDNS. This works similarly to the Multicast DNS Hybrid Proxy [10]. However, the DNSSD hybrid proxy exposes the topology of the network in which it operates to the user.

In order to avoid this, the homenet solution maintains a host namespace for each non-edge link in the homenet. Queries for names in the host namespace are looked up in the per-link host namespaces as well (and trigger mDNS queries as in the hybrid solution). When a cross-link name conflict is present for a name, the name is presented with a short modifier identifying the link.

For example, if two devices on two separate links both advertise the name 'janus' using mDNS, and the name 'janus' is not present in the host namespace, the two hosts' names are modified to, for example, 'janus-1' and 'janus-2'. If both devices present the human readable name 'Janus', then that name is presented as 'Janus (1)' and 'Janus (2)'. If the name 'janus' appears in the host namespace, then that name is presented just as 'janus'.

If a mDNS service advertises a name that appears in the host namespace, the HNR that hears the advertisement will defend the name, forcing the mDNS service to choose a different name.

This solution shares a problem that mdns hybrid has: user interfaces on hosts that present mDNS names in their mDNS format (e.g., 'janus.local') will not have a DNS entry for 'janus.local'. Connections to such hosts using the name presented in the UI will work when both hosts are attached to the same link, but not otherwise.

It is preferable that devices that are homenet-aware publish their names using DNS updates rather than using mDNS. mDNS is not supported as a query mechanism on homenets, other than in the sense that homeneds do not filter mDNS traffic on the local link. Service discovery is instead done using DNS service discovery [8]. This mechanism is supported on all modern devices that do service discovery, so there is no need to rely on mDNS.

### 3.4.2. DNS Update

DNS updates to the resolver on the local link are supported for adding names to local zones. When an update is received, if the name being updated does not exist, or if the update contains the same information as is present in the existing record, then the update is accepted. If a conflicting entry exists, the update is rejected.

This update procedure is available to hosts that implement DNS update for DNS service discovery, but are not homenet-aware. Hosts cannot delete records they have added, nor modify them; such records can only time out. Updates to server list records require that the host referenced by the update exist, and that the update come from that host. Such updates are additive, and are removed automatically when they become stale.

Hosts that are homenet-aware generate a KEY record containing a public key for which they retain the private key. They then publish their name in the host namespace, with whatever data they intend to publish on the name, and include the KEY record they have generated. The update is signed using SIG(0) on the provided key. If a record already exists, and does not contain the same KEY record, the update is refused. Otherwise it is accepted.

Homenet-aware hosts can then update their entries in the address table and in service tables by using their KEY record with SIG(0). Entries can be added and deleted. However, only modifications to RRs that reference the name in the host namespace are allowed; all other RRs must be left as they are.

### 3.5. Recovery from loss

In principle the names in the zone aren't precious. If there are multiple HNRs and one is replaced, the replacement recovers by copying the local namespaces and other info from the others. If all are lost, there are a few pieces of persistent data that need to be recovered:

- o The global name
- o The ZSK for both local namespaces
- o Names configured statically through the UI

All other names were acquired dynamically, and recovery is simply a matter of waiting for the device to re-announce its name, which will happen when the device is power cycled, and also may happen when it

sees a link state transition. The hybrid mDNS implementation will also discover devices automatically when service queries are made.

Devices that maintain their state using DNS update, but that are not homenet-aware, may or may not update their information when they see a link state transition. Homenet-aware devices will update whenever they see a link-state transition, and also update periodically. When the Homenet configuration has been lost, HNRs advertise a special ND option that indicates that naming and service discovery on the homenet is in a recovery state. Homenet-aware devices will be sensitive to this ND option, and will update when it is seen.

Homenets will present an standard management API, reachable through any homenet router, that allows a device that has stored the DNSSEC ZSK and KSK to re-upload it when it has been lost. This is safest solution for the end user: the keys can be stored on some device they control, under password protection.

ZSKs and KSKs can also be saved by the ISP or GNRP and re-installed using one of the management APIs. This solution is not preferable, since it means that the end user's security is reliant on the security of the GNRP or ISP's infrastructure.

If the ZSK and KSK are lost, they can be regenerated. This requires that the homenet's global name change: there is no secure way to re-key in this situation. Once the homenet has been renamed and re-keyed, all devices that use the homenet will simply see it as a different homenet.

### 3.6. Well-known names

Homenets serve a zone under the special-use top-level name [TBD2] that answers queries for local configuration information and can be used to advertise services provided by the homenet (as opposed to services present on the homenet). This provides a standard means for querying the homenet that can be assumed by management functions and homenet clients. A registry of well-known names for this zone is defined in IANA considerations (Section 9). Names and RRs in this zone are only ever provided by the homenet--this is not a general purpose service discovery zone.

All resolvers on the homenet will answer questions about names in this zone. Entries in the zone are guaranteed not to be globally unique: different homenets are guaranteed to give independent and usually different answers to queries against this zone. Hosts and services that use the special names under this TLD are assumed to be aware that it is a special TLD. If such hosts cache DNS entries, DNS



entries under this TLD are discarded whenever the host detects a network link state transition.

The `uuid.[TBD2]` name contains a TXT RR that contains the UUID of the homenet. Each homenet generates its own distinct UUID; homenet routers on any particular homenet all use the same UUID, which is agreed upon using HNCP. If the homenet has not yet generated a UUID, queries against this name will return NXDOMAIN.

The `global-name.[TBD2]` name contains a PTR record that contains the global name of the homenet. If the homenet does not have a global name, queries against this name will return NXDOMAIN.

The `global-name-register.[TBD2]` name contains one or more A and/or AAAA records referencing hosts (typically HNRs) that provide a RESTful API over HTTP that can be used to register the global name of the homenet, once that name has been configured.

The `all-resolver-names.[TBD2]` name contains an NS RRset listing a global name for each HNR. It will return NXDOMAIN if the homenet has no global name. These names are generated automatically by each HNR when joining the homenet, or when a homenet to which the HNR is connected establishes a global name.

#### 4. Name Resolution

##### 4.1. Configuring Resolvers

Hosts on the homenet receive a set of resolver IP addresses using either DHCP or RA. IPv4-only hosts will receive IPv4 addresses of resolvers, if available, over DHCP. IPv6-only hosts will receive resolver IPv6 addresses using either stateful (if available) or stateless DHCPv6, or through the domain name option in router advertisements. All homenet routers provide resolver information using both stateless DHCPv6 and RA; support for stateful DHCPv6 and DHCPv4 is optional, however if either service is offered, resolver addresses will be provided using that mechanism as well. Resolver IP addresses will always be IP addresses on the local link: every HNR is required to provide name resolution service. This is necessary to allow DNS update using presence on-link as a mechanism for rejecting off-network attacks.

##### 4.2. Configuring Service Discovery

DNS-SD uses several default domains for advertising local zones that are available for service discovery. These include the `'local'` domain, which is searched using mDNS, and also the IPv4 and IPv6 reverse zone corresponding to the prefixes in use on the local

network. For the homenet, no support for queries against the ".local" zone is provided by HNRs: a ".local" query will be satisfied or not by services present on the local link. This should not be an issue: all known implementations of DNSSD will do unicast queries using the DNS protocol.

Service discovery is configured using the technique described in Section 11 of DNS-Based Service Discovery [8]. HNRs will answer domain enumeration queries against every IPv4 address prefix advertised on a homenet link, and every IPv6 address prefix advertised on a homenet link, including prefixes derived from the homenet's ULA(s). Whenever the "<domain>" sequence appears in this section, it references each of the domains mentioned in this paragraph.

Homenets advertise the availability of several browsing zones in the "b.\_dns\_sd.<domain>" subdomain. The zones advertised are the "well known" zone (TBD2) and the zone containing the local namespace. If the global name is available, only that name is advertised for the local namespace; otherwise [TBD1] is advertised. Similarly, if the global name is available, it is advertised as the default browsing and service registration domain under "db.\_dns\_sd.<domain>", "r.\_dns\_sd.<domain>", "dr.\_dns\_sd.<domain>" and "lb.\_dns\_sd.<domain>"; otherwise, the name [TBD1] is advertised as the default.

#### 4.3. Resolution of local namespaces

The local namespace appears in two places, under [TBD1] and, if the homenet has a global name, under the global name. Resolution from inside the homenet yields the contents of the local namespaces; resolution outside of the homenet yields the contents of the public namespaces. If there is a global name for the homenet, RRs containing names in both instances of the local namespace are qualified with the global name; otherwise they are qualified with [TBD1].

#### 4.4. Service Discovery Resolution

Because homenets provide service discovery over DNS, rather than over mDNS, support for DNS push notifications [11]. When a query arrives for a local namespace, and no data exists in that namespace to answer the query, that query is retransmitted as an mDNS query. Data that exists to answer the query in mDNS cached namespaces does not prevent an mDNS query being issued.

If there is data available to answer the query in the host namespace or any of the dnssd cached namespaces, that data is aggregated and

returned immediately. If the host that sent the query requested push notification, then any mDNS responses that come in subsequent to the initial answer are sent as soon as they are received, and also added to the cache. This means that if a name has been published directly using DNS, no mDNS query for that name is ever generated.

#### 4.5. Local and Public Zones

The homenet's global name serves both as a unique identifier for the homenet and as a delegation point in the DNS for the zone containing the homenet's forward namespace. There are two versions of the forward namespace: the public version and the private version. Both of these versions of the namespace appear under the global name delegation, depending on which resolver a host is querying.

The homenet provides two versions of the zone. One is the public version, and one is the local version. The public version is never visible on the homenet (could be an exception for a guest net). The public version is available outside of the homenet. The local version is visible on the homenet. Whenever the zone is updated, it is signed with the ZSK. Both versions of the zone are signed; the local signed version always has a serial number greater than the public signed version. [we want to not re-sign the public zone if no public names in the private zone changed.]

This dual publication model relies on hosts connected to the homenet using the local resolver and not some external resolver. Hosts that use an external resolver will see the public version of the namespace. From a security UI design perspective, allowing queries from hosts on the homenet to resolvers off the homenet is risky, and should be prevented by default. This is because if the user sees inconsistent behavior on hosts that have external resolvers configured, they may attempt to fix this by making all local names public. If an alternate external resolver is to be used, it should be configured on the homenet, not on the individual host.

One way to make this work is to intercept all DNS queries to non-homenet IP addresses, check to see if they reference the local namespace, and if so resolve them locally, answering as if from the remote cache. If the query does not reference a local namespace, and is listed as "do not forward" in RFC 6761 or elsewhere, it can be sent to the intended cache server for resolution without any special handling for the response. This functionality is not required for homenet routers, but is likely to present a better user experience.

#### 4.6. DNSSEC Validation

All namespaces are signed using the same ZSK. The ZSK is signed by a KSK, which is ideally kept offline. Validation for the global name is done using the normal DNSSEC trust hierarchy. Validation for the [TBD1] and [TBD2] zones can be done by fetching the global name from the [TBD2] zone, fetching and validating the ZSK using DNSSEC, and then using that as a trust anchor.

Only homenet-aware hosts will be able to validate names in the [TBD1] and [TBD2] zones. The homenet-aware host validates non-global zones by determining which homenet it is connected to querying the uuid.[TBD2] and global-name.[TBD2] names. If there is an answer for the global-name.[TBD2] query, validation can proceed using the trust anchor published in the zone that delegates the global name. If only the uuid is present, then the homenet-aware host can use trust-on-first-use to validate that an answer came from the homenet that presented that UUID. This provides only a limited degree of trustworthiness.

#### 4.7. Support for Multiple Provisioning Domains

Homenets must support the Multiple Provisioning Domain Architecture [9]. In order to support this architecture, each homenet router that provides name resolution must provide one resolver for each provisioning domain (PvD). Each homenet router will advertise one resolver IP address for each PvD. DNS requests to the resolver associated with a particular PvD, e.g. using RA options [12] will be resolved using the external resolver(s) provisioned by the service provider responsible for that PvD.

The homenet is a separate provisioning domain from any of the service providers. The global name of the homenet can be used as a provisioning domain identifier, if one is configured. Homenets should allow the name of the local provisioning domain to be configured; otherwise by default it should be "Home Network xxx", where xxx is the generated portion of the homenet's ULA prefix, represented as a base64 string.

The resolver for the homenet PvD is offered as the primary resolver in RAs and through DHCPv4 and DHCPv6. When queries are made to the homenet-PvD-specific resolver for names that are not local to the homenet, the resolver will use a round-robin technique, alternating between service providers with each step in the round-robin process, and then also between external resolvers at a particular service provider if a service provider provides more than one. The round-robinning should be done in such a way that no service provider is preferred, so if service provider A provides one caching resolver

(A), and service provider B provides two (B1, B2), the round robin order will be (A, B1, A, B2), not (A, B1, B2).

Every resolver provided by the homenet, regardless of which provisioning domain it is intended to serve, will accept updates for services in the local service namespace from hosts on the local link.

#### 4.8. Using the Local Namespace While Away From Home

Homenet routers do not answer unauthenticated DNS queries from off the local network. However, some applications may benefit from the ability to resolve names in the local namespace while off-network. Therefore hosts connected to the homenet can register keys in the host namespace using DNS Update. Such keys must be validated by the end user before queries against the local namespace can be authenticated using that key. A host that will make remote queries to the local namespace caches the names of all DNS servers on the homenet by querying all-resolver-names.[TBD2].

Hosts that require name resolution from the local network must have a stub resolver configured to contact the dns server on one or more routers in the homenet when resolving names in the host or address namespaces. To do this, resolvers must know the global name of the local namespace, which they can retain from previous connections to the homenet.

The homenet may not have a stable IP address, so such resolvers cannot merely cache the IP address of the homenet routers. Instead, they cache the NS record listing the HNRs and use those names to determine the IP addresses of the homenet routers at the time of resolution. Such IP addresses can be safely cached for the duration of the TTL of the A or AAAA record that contained them. The names of the homenet router DNS servers should be randomly generated so that they can't be guessed by off-network attackers.

To make a homenet DNS query, the host signs the request using SIG(0) with the key that they registered to the homenet. The homenet router first checks the question in the query for validity: it must be a subdomain of the global name. The homenet router then checks the name of the signing key against the list of cached, validated keys; if that key is cached and validated, then the homenet router attempts to validate the SIG(0) signature using that key. If the signature is valid, then the homenet router answers the query. If the zone doesn't have a trust anchor in the parent zone, the responding server signs the answer with its own ZSK. The resolver that sent the query validates the response using DNSSEC if possible, and otherwise using the ZSK directly.

## 5. Publishing the Public Namespace

### 5.1. Acquiring the Global Name

There are two ways to acquire a global name: the end-user can register a domain name using a public domain name registry, or the end-user can be assigned a subdomain of a registered domain by a homenet global name service provider. We will refer to this as the Global Name Registration Provider [GNRP]. In either case, the registration process can either be manual or automatic. Homenet routers support automatic registration regardless of the source of the homenet's global name, using a RESTful API.

### 5.2. Hidden Primary/Public Secondaries

The default configuration for a homenet's external name service is that the primary server for the zone is not published in an NS record in the zone's delegation. Instead, the GNRP provides authoritative name service for the zone. Whenever the public zone is updated, the hidden primary sends NOTIFY messages to all the secondaries, using the zone's ZSK to sign the message.

When any of the GNRP secondary servers receives a notify for the zone, it checks to see that the notify is signed with a valid ZSK for that zone. If so, it contacts the IP address from which the NOTIFY was sent and initiates a zone transfer. Using this IP address avoids renumbering issues. Upon finishing the zone transfer, the zone is validated using each ZSK used to sign it. If any validation fails, the new version of the zone is discarded. If updates have been received, but no valid updates received, over a user-settable interval defaulting to a day (or?), the GNRP will communicate to the registered user that there is a problem.

The reverse zone for any prefix delegated by an ISP should be delegated by that ISP to the home gateway to which the delegation was sent. The list of secondaries for that zone is sent to the home gateway using DHCPv6 prefix delegation. The ZSK is announced to the ISP in each DHCP PD message sent by the home gateway. Whenever an update is made to this zone, the home gateway sends a NOTIFY to each of the listed secondaries for the delegation, and updates occur as described above. Once the delegation is established, the ISP will not accept a different ZSK unless the prefix and its delegated zone are reassigned.

### 5.3. PKI security

All communication with the homenet using HTTP is encrypted using opportunistic security. If the homenet is configured with PKI, then the PKI certificate is used. Homenets should automatically acquire a PKI certificate when a global name is established. This certificate should be published in a TLSA record in the host namespace on any hostnames on which HTTP service is offered by HNRs.

### 5.4. Renumbering

The homenet may renumber at any time. IP address RRs published in any namespace must never have a TTL that is longer than the valid lifetime for the prefix from which the IP address was allocated. If a particular ISP has deprecated a prefix (its preferred lifetime is zero), IP addresses derived from that prefix are not published in the any namespace. If more than one prefix is provided by the same ISP and some have different valid lifetimes, only IP addresses in the prefix or prefixes with the longest valid lifetime are published.

### 5.5. ULA

Homenets have at least one ULA prefix. If a homenet has two ULA prefixes, and one is deprecated, addresses in the second ULA prefix are not published. The default source address selection algorithm ensures that if a service is available on a ULA, that ULA will be used rather than the global address. Therefore, no special effort is made in the DNS to offer only ULAs in response to local queries.

## 6. Management

### 6.1. End-user management

Homenets provide two management mechanisms for end users: an HTTP-based user interface and an HTTP-based RESTful API [tbw].

Homenets also provide a notification for end users. By default, when an event occurs that requires user attention, the homenet will attract the user's attention by triggering captive portal detection on user devices. Users can also configure specific devices to receive management alerts using the RESTful management API; in this case, no captive portal notification is performed.

### 6.2. Central management

Possibly can be done mostly through RESTful API, but might want Netconf/Yang as well. Should be possible to have the local namespace mastered on an external DNS auth server, e.g. in case a bunch of HNRs

are actually set up in an org, or in case an ISP wants to provide a service package for users who would rather not have an entirely self-operating network.

## 7. Privacy Considerations

Private information must not leak out as a result of publishing the public namespace. The 'public' flag on RRsets in homenet-managed namespaces prevents leakage of information that has not been explicitly marked for publication.

The privacy of host information on the local net is left to hosts. Various mechanisms are available to hosts to ensure that tracking does not occur if it is not desired. However, devices that need to have special permission to manage the homenet will inevitably reveal something about themselves when doing so. It may be possible to use something like HTTP token binding[13] to mitigate this risk.

## 8. Security Considerations

There are some clear issues with the security model described in this document, which will be documented in a future version of this section. A full analysis of the avenues of attack for the security model presented here have not yet been done, and must be done before the document is published.

## 9. IANA considerations

IANA will add a new registry titled Homenet Management Well-Known Names, which initially contains:

uuid Universally Unique Identifier--TXT record containing, in base64 encoding, a stable, randomly generated identifier for the homenet that is statistically unlikely to be shared by any other homenet.

global-name The homenet's global name, represented as a PTR record to that name.

global-name-register The hostname of the homenet's global name registry service, with A and/or AAAA records.

all-resolver-names A list of all the names of the homenet's resolvers for the homenet Pvd, represented as an RRset containing one or more PTR records.

The IANA will allocate two names out of the Special-Use Domain Names registry:



TBD1 Suggested value: "homenet"

TBD2 Suggested value: "\_hnsd"

## 10. Normative References

- [1] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <<http://www.rfc-editor.org/info/rfc1034>>.
- [2] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <<http://www.rfc-editor.org/info/rfc1035>>.
- [3] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, DOI 10.17487/RFC1918, February 1996, <<http://www.rfc-editor.org/info/rfc1918>>.
- [4] Vixie, P., Ed., Thomson, S., Rekhter, Y., and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)", RFC 2136, DOI 10.17487/RFC2136, April 1997, <<http://www.rfc-editor.org/info/rfc2136>>.
- [5] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, DOI 10.17487/RFC4193, October 2005, <<http://www.rfc-editor.org/info/rfc4193>>.
- [6] Andrews, M., "Locally Served DNS Zones", BCP 163, RFC 6303, DOI 10.17487/RFC6303, July 2011, <<http://www.rfc-editor.org/info/rfc6303>>.
- [7] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, DOI 10.17487/RFC6762, February 2013, <<http://www.rfc-editor.org/info/rfc6762>>.
- [8] Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", RFC 6763, DOI 10.17487/RFC6763, February 2013, <<http://www.rfc-editor.org/info/rfc6763>>.
- [9] Anipko, D., Ed., "Multiple Provisioning Domain Architecture", RFC 7556, DOI 10.17487/RFC7556, June 2015, <<http://www.rfc-editor.org/info/rfc7556>>.
- [10] Cheshire, S., "Hybrid Unicast/Multicast DNS-Based Service Discovery", draft-ietf-dnssd-hybrid-03 (work in progress), February 2016.

- [11] Pusateri, T. and S. Cheshire, "DNS Push Notifications", draft-ietf-dnssd-push-07 (work in progress), April 2016.
- [12] Korhonen, J., Krishnan, S., and S. Gundavelli, "Support for multiple provisioning domains in IPv6 Neighbor Discovery Protocol", draft-ietf-mif-mpvd-ndp-support-03 (work in progress), February 2016.
- [13] Popov, A., Nystrom, M., Balfanz, D., Langley, A., and J. Hodges, "Token Binding over HTTP", draft-ietf-tokbind-https-05 (work in progress), July 2016.

Author's Address

Ted Lemon  
Nominum, Inc.  
800 Bridge Parkway  
Redwood City, California 94065  
United States of America

Phone: +1 650 381 6000  
Email: ted.lemon@nominum.com

Internet Engineering Task Force  
Internet-Draft  
Intended status: Informational  
Expires: January 3, 2018

S. Cheshire  
Apple Inc.  
T. Lemon  
Nominum, Inc.  
July 2, 2017

Service Discovery Broker  
draft-sctl-discovery-broker-00

Abstract

DNS-Based Service Discovery allows clients to discover available services using unicast DNS queries. In simple configurations these unicast DNS queries go directly to the appropriate authoritative server(s). In large networks that have complicated topology, or many client devices, or both, it can be advantageous to have an intermediary between the clients and authoritative servers. This intermediary, called a Discovery Broker, serves several purposes. A Discovery Broker can reduce load on both the servers and the clients, and gives the option of presenting clients with service discovery organized around logical, rather than physical, topology.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on January 3, 2018.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## 1. Introduction

DNS-Based Service Discovery (DNS-SD) [RFC6763] is a component of Zero Configuration Networking [RFC6760] [ZC] [Roadmap].

DNS-SD operates on a single network link (broadcast domain) using Multicast DNS [RFC6762]. DNS-SD can span multiple links using unicast DNS.

In the DNS-SD specification [RFC6763] section 11, "Discovery of Browsing and Registration Domains (Domain Enumeration)", describes how client devices are automatically configured with the appropriate unicast DNS domains in which to perform their service discovery queries. When used in conjunction with a Discovery Proxy [DisProx] this allows clients to discover services on remote links, even when the devices providing those services support only the basic Multicast DNS form of DNS-Based Service Discovery. A Discovery Broker is a companion technology that operates in conjunction with existing authoritative DNS servers (such as a Discovery Proxy [DisProx]) and existing clients performing service discovery using unicast DNS queries.

## 2. Problem Statement

The following description of how a Discovery Broker works is illustrated using the example of a long rectangular office building. The building is large enough to have hundreds or even thousands of employees working there, the network is large enough that it would be impractical to operate it as a single link (a single broadcast domain, with a single IPv4 subnet or IPv6 network prefix).

Suppose, for this example, that the network is divided into twelve separate links, connected by routers. Each link has its own IPv6 network prefix. The division of the network into twelve sections of roughly equal size is somewhat arbitrary, and does not necessarily follow any physical boundaries in the building that are readily apparent to its inhabitants. Two people in adjacent offices on the same corridor may have Ethernet ports connected to different links. Indeed, two devices in the same office, connected to the company network using secure Wi-Fi, may inadvertently associate with different access points, which happen to be connected to different wired links with different IPv6 network prefixes.

If this network were operated the way most networks have historically been operated, it would use only Multicast DNS Service Discovery, and adjacent devices that happen to connect to different underlying links would be unable to discover each other. And this would not be a rare occurrence. Since this example building contains eleven invisible boundaries between the twelve different links, anyone close to one of those invisible boundaries will have a population of nearby devices that are not discoverable on the network, because they're on a different link. For example, a shared printer in a corridor outside one person's office may not be discoverable by the person in the very next office.

One path to solving this problem is as follows:

1. Install a Discovery Proxy [DisProx] on each of the twelve links.
2. Create twelve named subdomains, such as, "services1.example.com", "services2.example.com", "services3.example.com", and so on.
3. Delegate each named subdomain to the corresponding Discovery Proxy on that link.
4. Create entries in the 'ip6.arpa' reverse mapping zone directing clients on each link to perform service discovery queries in the appropriate named subdomains, as documented in section 11 of the DNS-SD specification [RFC6763].

In step 4 above, it might be tempting to add only a single record in each reverse mapping domain referencing the corresponding services subdomain. This would work, but it would only facilitate each client discovering the same services it could already discover using Multicast DNS [RFC6762]. In some cases even this is useful, such as when using Wi-Fi Access Points with multicast disabled for efficiency. In such cases this configuration would allow wireless clients to discover services on the wired network segment without having to use costly Wi-Fi multicast.

But for this example we want to achieve more than just equivalency with Multicast DNS.

In this example, each reverse mapping domain is populated with the name of its own services subdomain, plus its neighbors. The reverse mapping domain for the first link has two "lb.\_dns-sd.\_udp" PTR records, referencing "services1.example.com" and "services2.example.com". The second link references services1, services2, and services3. The third link references services2, services3, and services4. This continues along the building, until the last link, which references services11 and services12.

In this way a "sliding window" is created, where devices on each link are directed to look for services both on that link and on its two immediate neighbors. Depending on the physical and logical topologies of the building and its network, it may be appropriate to direct clients to query in more than three services subdomains. If the building were a ring instead of a linear rectangle, then the network topology would "wrap around", so that links 1 and 12 would be neighbors.

This solves the problem of being unable to discover a nearby device because it happens to be just the other side of one of the twelve arbitrary invisible network link boundaries.

For many cases this solution is adequate, but there is an issue to consider. In the example above, a client device on link 5 has TCP connections to three Discovery Proxies, on links 4, 5 and 6. In a more complex setup each client could have many more TCP connections to different Discovery Proxies.

Similarly, if there are a many clients, each Discovery Proxy could be required to handle thousands of simultaneous TCP connections from clients.

The solution to these two problems is the Discovery Broker.

### 3. Discovery Broker Operation

The Discovery Broker is an intermediary between the client devices and the Discovery Proxies. It is a kind of multiplexing crossbar switch. It shields the clients from having to connect to multiple Discovery Proxies, and it shields the Discovery Proxies from having to accept connections from thousands of clients.

Each client needs only a single TCP connection to a single Discovery Broker, rather than multiple TCP connections directly to multiple Discovery Proxies. This eases the load on client devices, which may be mobile and battery-powered.

Each Discovery Proxy needs to support connections to at most a small number of Discovery Brokers. The burden of supporting thousands of clients is taken by the Discovery Broker, which can be a powerful server in a data center. This eases the load on the Discovery Proxy, which may be implemented in a device with limited RAM and CPU resources, like a Wi-Fi access point or IP router.

Recall that a Discovery Proxy [DisProx] is a special kind of authoritative DNS server [RFC1034] [RFC1035]. Externally it behaves like a traditional authoritative DNS server, except that instead of getting its zone data from a manually-administered zone file, it learns its zone data dynamically as a result of performing Multicast DNS queries on its local link.

A Discovery Broker is a similar concept, except that it learns its zone data dynamically as a result of performing \*unicast\* DNS queries. For example, a Discovery Broker could be configured so that the answer for "<something>.discovery5.example.com" is obtained by performing corresponding unicast DNS queries:

```
<something>.services4.example.com
<something>.services5.example.com
<something>.services6.example.com
```

and then returning the union of the results as the answer. The rdata of the returned answers is not rewritten or modified in any way by the Discovery Broker.

#### 4. Protocol Transparency

From the point of view of an authoritative DNS server such as a Discovery Proxy, the protocol a Discovery Broker uses to make requests of it is the exact same DNS protocol that any other client would use to make requests of it (which may be traditional one-shot DNS queries [RFC1034] [RFC1035] or long-lived DNS Push Notifications [Push]).

A Discovery Broker making requests is no different from any other client making requests. The fact that the Discovery Broker may be making a single request on behalf of thousands of clients making the same request, thereby shielding the Discovery Proxy from excessive traffic burden, is invisible to the Discovery Proxy.

This means that an authoritative DNS server such as a Discovery Proxy does not have to be aware that it is being queried by a Discovery Broker. In some scenarios a Discovery Proxy may be deployed with clients talking to it directly; in other scenarios the same Discovery Proxy product may be deployed with clients talking via a Discovery Broker. The Discovery Proxy simply answers queries as usual in both cases.

Similarly, from the point of view of a client, the protocol it uses to talk to a Discovery Broker is the exact same DNS protocol it uses to talk to a Discovery Proxy or any other authoritative DNS server.

This means that the client does not have to be aware that it is using a Discovery Broker. The client simply sends service discovery queries as usual, according to configuration it received from the network or otherwise, and receives answers as usual. A Discovery Broker may be employed to shield a Discovery Proxy from excessive traffic burden, but this is transparent to a client.

Another benefit for the client is that by having the Discovery Broker query multiple subdomains and aggregate the results, it saves the client from having to do multiple separate queries of its own.



## 5. Logical vs. Physical Topology

In the example so far, we have focussed on facilitating discovery of devices and services that are physically nearby.

Another application of the Discovery Broker is to facilitate discovery of devices and services according to other logical relationships.

For example, it might be considered desirable for the company's two file servers to be discoverable company-wide, but for its many printers to only be discovered (by default) by devices on nearby network links.

As another example, company policy may block access to certain resources from Wi-Fi; in such cases it would make sense to implement consistent policies at the service discovery layer, to avoid the user frustration of services being discoverable on Wi-Fi that are not usable from Wi-Fi.

Such policies, and countless variations thereon, may be implemented in a Discovery Broker, limited only by the imagination of the vendor creating the Discovery Broker implementation.

## 6. Recursive Application

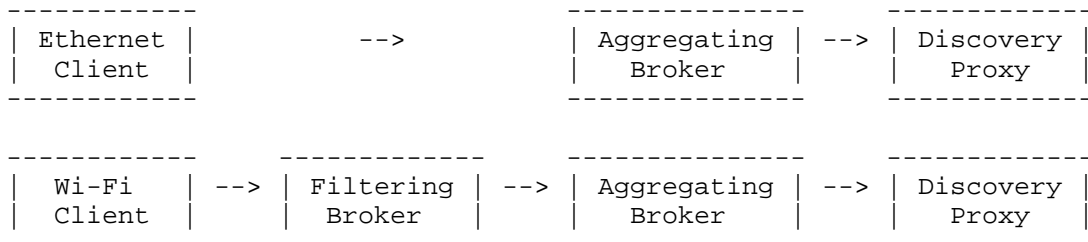
Due to the Protocol Transparency property described above, multiple Discovery Brokers may be "stacked" in whatever combinations are useful. A Discovery Broker makes queries in exactly the same way a client would, and a Discovery Broker accepts queries in exactly the same way a Discovery Proxy (or other authoritative DNS server) would. This means that a Discovery Broker talking to another Discovery Broker is no different from client-to-broker or broker-to-proxy communication, or indeed, direct client-to-proxy communication. The arrows in the chart below are all instances of the same communication protocol.

client -> proxy

client -> broker -> proxy

client -> broker -> broker -> proxy

This makes it possible to combine Discovery Brokers with different functionality. A Discovery Broker performing physical aggregation could be used in conjunction with a Discovery Broker performing policy-based filtering, as illustrated below:



## 7. Security Considerations

Discovery (or non-discovery) of services is not a substitute for suitable access control. Servers listening on open ports are generally discoverable via a brute-force port scan anyway; DNS-Based Service Discovery makes access to these services easier for legitimate users.

## 8. Informative References

- [RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <<http://www.rfc-editor.org/info/rfc1034>>.
- [RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <<http://www.rfc-editor.org/info/rfc1035>>.
- [RFC6760] Cheshire, S. and M. Krochmal, "Requirements for a Protocol to Replace the AppleTalk Name Binding Protocol (NBP)", RFC 6760, DOI 10.17487/RFC6760, February 2013, <<http://www.rfc-editor.org/info/rfc6760>>.
- [RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, DOI 10.17487/RFC6762, February 2013, <<http://www.rfc-editor.org/info/rfc6762>>.
- [RFC6763] Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", RFC 6763, DOI 10.17487/RFC6763, February 2013, <<http://www.rfc-editor.org/info/rfc6763>>.
- [Roadmap] Cheshire, S., "Service Discovery Road Map", draft-cheshire-dnssd-roadmap-00 (work in progress), July 2017.
- [DisProx] Cheshire, S., "Discovery Proxy for Multicast DNS-Based Service Discovery", draft-ietf-dnssd-hybrid-06 (work in progress), March 2017.
- [Push] Pusateri, T. and S. Cheshire, "DNS Push Notifications", draft-ietf-dnssd-push-12 (work in progress), July 2017.
- [ZC] Cheshire, S. and D. Steinberg, "Zero Configuration Networking: The Definitive Guide", O'Reilly Media, Inc. , ISBN 0-596-10100-7, December 2005.

Authors' Addresses

Stuart Cheshire  
Apple Inc.  
1 Infinite Loop  
Cupertino, California 95014  
USA

Phone: +1 408 974 3207  
Email: cheshire@apple.com

Ted Lemon  
Nominum, Inc.  
800 Bridge Parkway  
Redwood City, California 94065  
United States of America

Phone: +1 650 381 6000  
Email: ted.lemon@nominum.com

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: January 4, 2018

S. Cheshire  
Apple Inc.  
T. Lemon  
Nominum, Inc.  
July 3, 2017

Multicast DNS Discovery Relay  
draft-sctl-dnssd-mdns-relay-00

Abstract

This document extends the Discovery Proxy for Multicast DNS-Based Service Discovery specification. It describes a lightweight relay mechanism, a Discovery Relay, which allows Discovery Proxies to provide service on links to which the hosts on which they are running are not directly attached.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on January 4, 2018.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
2. Terminology . . . . .	4
3. Protocol Overview . . . . .	5
3.1. Connections between Discovery and Discovery Relays . . . . .	5
3.2. mDNS Messages On Links . . . . .	6
4. Orchestration . . . . .	6
4.1. Orchestration System Functional Overview . . . . .	7
4.2. Orchestration Primitives . . . . .	7
4.2.1. Link Present . . . . .	7
4.2.2. Link Remove . . . . .	8
4.2.3. Discovery Proxy Available . . . . .	8
4.2.4. Discovery Proxy Resigning . . . . .	8
4.2.5. Discovery Relay Available . . . . .	9
4.2.6. Discovery Relay Resigning . . . . .	9
4.3. Orchestration System Behavior . . . . .	10
4.3.1. Link Present . . . . .	10
4.3.2. Link Remove . . . . .	10
4.3.3. Discovery Proxy Available . . . . .	10
4.3.4. Discovery Proxy Resigning . . . . .	11
4.3.5. Discovery Relay Available . . . . .	11
4.3.6. Discovery Relay Resigning . . . . .	12
4.3.7. Node Available . . . . .	12
4.3.8. Node Resigning . . . . .	12
4.4. Discovery Proxy Performer Behavior . . . . .	12
4.4.1. Link Present . . . . .	13
4.4.2. Link Removed . . . . .	13
4.4.3. Discovery Proxy Available . . . . .	13
4.4.4. Discovery Proxy Resigning . . . . .	14
4.4.5. Discovery Relay Available . . . . .	14
4.4.6. Discovery Relay Resigning . . . . .	15
4.5. Discovery Relay Performer Behavior . . . . .	15
4.5.1. Link Present . . . . .	15
4.5.2. Link Removed . . . . .	15
4.5.3. Discovery Proxy Available . . . . .	15
4.5.4. Discovery Proxy Resigning . . . . .	16
4.5.5. Discovery Relay Available . . . . .	16
4.5.6. Discovery Relay Resigning . . . . .	16
5. Connections between Discovery Proxies and Discovery Relays . . . . .	16
6. Traffic from Relays to Proxies . . . . .	18
7. Traffic from Proxies to Relays . . . . .	19
8. Discovery Proxy Behavior . . . . .	19
9. Session Signaling TLVs . . . . .	19
9.1. MDNS Link Request . . . . .	19

9.2.	MDNS Link Invalid . . . . .	19
9.3.	MDNS Link Subscribed . . . . .	20
9.4.	MDNS Message . . . . .	20
9.5.	Layer 2 Source Address . . . . .	20
9.6.	IP Source Address . . . . .	20
9.7.	IP Source Port . . . . .	21
9.8.	Link Identifier . . . . .	21
9.9.	MDNS Discontinue . . . . .	21
9.10.	IP Address Family . . . . .	21
10.	Security Considerations . . . . .	21
11.	IANA Considerations . . . . .	21
12.	IANA Considerations . . . . .	22
13.	Acknowledgments . . . . .	22
14.	References . . . . .	22
14.1.	Normative References . . . . .	22
14.2.	Informative References . . . . .	23
	Authors' Addresses . . . . .	23

## 1. Introduction

The Discovery Proxy for Multicast DNS-Based Service Discovery [I-D.ietf-dnssd-hybrid] specification defines a mechanism for discovering services on a subnetted network using Multicast DNS (mDNS) [RFC6762], through the use of Discovery Proxies, which issue mDNS requests on various links in the network on behalf of a host attempting service discovery.

In the original Discovery Proxy specification, it is assumed that for every link on which services will be discovered, a host will be present running a full Discovery Proxy. This document introduces a lightweight Discovery Relay which can be used to provide discovery services on a link without requiring a full Discovery Proxy on every link.

The Discovery Relay operates by listening for TCP connections from Discovery Proxies. When a Discovery Proxy connects, the connection is authenticated and secured using TLS. The Discovery Proxy can then send messages that will be relayed to specified links. The Discovery Proxy may also specify one or more links from which it wishes to receive mDNS traffic. DNS Session Signaling [I-D.ietf-dnsop-session-signal] is used as a framework for conveying interface and IP header information associated with each message.

The Discovery Relay functions essentially as a set of one or more virtual interfaces for the Discovery proxy, one on each link to which the Discovery Relay is connected. In a complex network, it is possible that more than one Discovery Relay will be connected to the same link; in this case, the Discovery Proxy ideally should only be

using one such Relay Proxy per link, since using more than one will generate duplicate traffic.

How such duplication is detected and avoided is out of scope for this document: in principle it could be detected using HNCP [RFC7788] or configured using some sort of orchestration software in conjunction with NETCONF [RFC6241] or CPE WAN Management Protocol [TR-069].

## 2. Terminology

The following definitions may be of use:

**mDNS Agent** A host which sends and/or responds to mDNS queries.

**Discovery Proxy** A network service which receives well-formed questions using the DNS protocol, performs multicast DNS queries to answer those questions, and responds with those answers using the DNS protocol.

**Discovery Relay** A network service which sends mDNS messages on behalf of a Discovery Proxy and relays mDNS messages to a Discovery Relay.

**link** A maximal set of network connection points such that any host connected to any connection point may send a packet to a host connected to any other connection point without the help of a layer 3 router.

**whitelist** A list of one or more IP addresses from which a Discovery Relay may accept connections.

**silently discard** When a message that is not supported or not permitted is received, and the required response to that message is to "silently discard" it, that means that no response is sent by the service that is discarding the message to the service that sent it. The service receiving the message may log the event, and may also count such events: "silently" does not preclude such behavior.

**Director** A central or coordinated controlling function in an orchestrated network of Discovery Proxies and Discovery Relays (Section 4.1).

**Performer** The interface through which the Director directs the behavior of Discovery Proxies and Discovery Relays (Section 4.1).



### 3. Protocol Overview

This document describes a way for Discovery Proxies to communicate with mDNS agents on networks to which they are not directly connected using a Discovery Relay. As such, there are two parts to the protocol: connections between Discovery Proxies and Discovery Relays, and communications between Discovery Relays and mDNS agents.

#### 3.1. Connections between Discovery and Discovery Relays

Discovery Relays listen for connections. Connections between Discovery Proxies and Discovery Relays are established by Discovery Proxies. Connections are authenticated and encrypted using TLS, with both client and server certificates. Connections are long-lived: a Discovery Proxy is expected to send many queries over the same connection, and Discovery Relays will forward all mDNS traffic from subscribed interfaces over the connection.

The stream encapsulated in TLS will carry DNS frames as in the DNS TCP protocol [RFC1035] Section 4.2.2. However, all messages will be DNS Session Signaling messages [I-D.ietf-dnsop-session-signal]. There will be three types of such messages:

- o Subscribe messages from Discovery Proxy to Discovery Relay
- o mDNS messages from Discovery Proxy to Discovery Relay
- o mDNS messages from Discovery Relay to Discovery Proxy

Subscribe messages from the Discovery Proxy to the Discovery Relay indicate to the Discovery Relay that mDNS messages from one or more specified links are to be relayed to the Discovery Proxy.

mDNS messages from a Discovery Proxy to a Discovery Relay cause the Discovery Relay to re-transmit the mDNS message on one or more links to which the Discovery Relay host is directly attached.

mDNS messages from a Discovery Relay to a Discovery Proxy are sent whenever an mDNS message is received on a link to which the Discovery Relay has subscribed.

Discovery Relays are responsible for keeping connections alive when no traffic has been sent during a keepalive period [I-D.ietf-dnsop-session-signal] Section 4.

### 3.2. mDNS Messages On Links

Discovery Relays listen for mDNS traffic on all configured links. When a mDNS message is received on a link, it is forwarded on every open Discovery Proxy connection that is subscribed to mDNS traffic on that link. In the event of congestion, where a particular Discovery Proxy connection has no buffer space for an mDNS message that would otherwise be forwarded to it, the mDNS message is not forwarded to it. Normal mDNS retry behavior is used to recover from this sort of packet loss. Discovery Relays are not expected to buffer more than a few mDNS packets.

Discovery Relays accept mDNS traffic from Discovery Proxies. Such traffic is forwarded to zero or more links to which the Discovery Relay host is directly connected.

## 4. Orchestration

In order for one or more Discovery Proxies to make use of one or more Discovery Relays to provide service discovery on one or more links, the set of links on which service will be provided must be known, the set of Discovery Relays for those links must be known, and the set of Discovery Proxies allowed to connect to those Discovery Relays must be known. We assume that this information is maintained in some sort of orchestration system.

Although it is of course possible to configure such an environment with a set of static configuration files, it is most useful to consider such a network to be dynamic, with links potentially being added and removed, Discovery Proxies being added and removed, and Discovery Relays being added and removed. This document takes no position on which specific orchestration system will be used, but does specify the inputs and outputs of such a system that will be required for successful operation. In the case of static configuration, these inputs and outputs are also the same; the only difference is that they do not change without human intervention.

It is not strictly necessary that all participants in the orchestration process have complete information. It may be desirable for example to have more than one Discovery Proxy managed by an orchestration system, but to have different Discovery Proxies support different links. The set of primitives described here can be used to implement configurations where multiple Discovery Proxies are present and supporting disjoint, overlapping or identical sets of links.

There is a special case of orchestration that may be desirable in some settings: when a node may need to be capable of providing either Discovery Proxy service or Discovery Relay service, and is configured

to provide Discovery Proxy service, it would be useful to have a way to automatically configure the Discovery Relay to use the Discovery Proxy just on that one node, without requiring a network-wide orchestration system. In the case of a node that supports orchestration through HNCP, however, this is unnecessary: HNCP will work to provide orchestration even on a single node.

#### 4.1. Orchestration System Functional Overview

Conceptually, the orchestration system has two parts: the part that manages the network, and the part an instance of which is present on each node in the network that is orchestrated by the system. In a cooperative system such as HNCP [RFC7788], orchestration is done cooperatively, and the two functions are present on every participating node. In a managed system using NETCONF [RFC6241], a central service pushes configuration information to managed nodes, and pulls status information from managed nodes. For this discussion, which of these models is used (or whether some other model is used) is immaterial. The functional division is the same in either case: conceptually there is one function that does the orchestration called the Director, and there are one or more functions to which the orchestration applies, called Performers.

The Director is receptive to primitives from Performers. Performers apply primitives announced to them by the Director, and announce primitives to the Director. The Director announces primitives to Performers, based on its operating model and its configuration, based either in changes to the network or to announcements from Performers.

It is permissible for nodes to provide both Discovery Proxy and Discovery Relay service at the same time. In this case, there is a further conceptual functional division: on such a node, there are two Performers: the Discovery Proxy Performer and the Discovery Relay Performer. These may be the same program, or they may be functionally separate; which is the case is beyond the scope of this document. The reason for making this distinction is to point out that on a node providing both services, both Performers may receive every announcement sent by the Director. And of course the Director receives announcements sent by either Performer.

#### 4.2. Orchestration Primitives

##### 4.2.1. Link Present

The 'Link Present' primitive is used by the Director to communicate the presence of a link to Performers. 'Link Present' primitives include the following data:

**link identifier** One or more opaque 32-bit identifiers, each of which identifies a link that is present on the orchestrated network. Each identifier is unique among all link identifiers managed by the Director. These link identifiers are used in the Discovery Relay protocol to identify links on which mDNS requests will be sent and received, and are consistent across all participants in the orchestration system.

#### 4.2.2. Link Remove

The 'Link Remove' primitive is used by the Director to communicate to Performers that a link that was formerly present is no longer present. The 'Link Remove' primitive includes the following data:

**link identifier** One or more opaque 32-bit identifiers, as described in Section 4.2.1.

#### 4.2.3. Discovery Proxy Available

The 'Discovery Proxy Available' primitive is used by Discovery Proxy Performers to announce their availability to the Director, and by the Director to announce to Discovery Relay Performers that Discovery Proxies are present and enabled. This primitive is only used for nodes that provide Discovery Proxy service and can use Discovery Relays: a Discovery Proxy that does not support Discovery Proxy service is never announced in this way. The 'Discovery Proxy Available' primitive includes the following data:

**node identifier** The node identifier of the Discovery Proxy, unique among all nodes managed by a Director.

**IP addresses** One or more IP addresses configured on the network interfaces of the node making the announcement. This list must include all IP addresses from which the Discovery Proxy might connect to Discovery Relays, but need not include any other IP addresses.

**TLS Certificate** A TLS PKI certificate or bare public key which will be used by the Discovery Proxy to authenticate itself when connecting to Discovery Relays.

#### 4.2.4. Discovery Proxy Resigning

The 'Discovery Proxy Resigning' primitive is used by Discovery Proxies to announce to the Director that they are no longer available, and by the Director to announce to Discovery Relay performers that a Discovery Proxy is no longer present or enabled.

The 'Discovery Proxy Resigning' primitive includes the following data:

node identifier The node identifier of the Discovery Proxy, unique among all nodes managed by a Director.

#### 4.2.5. Discovery Relay Available

The 'Discovery Relay Available' primitive is used by Discovery Relay Performers to inform the Director that they are available to provide service. It is used by the Director to announce to Discovery Proxy Performers that a Discovery Relay is available and enabled. The 'Discovery Relay Available' primitive includes the following data:

node identifier The node identifier of the Discovery Relay, unique among all nodes managed by a Director.

IP addresses A list of IP addresses on which the Discovery Relay may be contacted.

Port TCP Port on which the Discovery Relay will be listening for connections.

Server Certificate A TLS PKI certificate or bare public key which will be presented to Discovery Proxies when they initiate TLS connections with the Discovery Relay. This is used both to authenticate the Discovery Relay, and also to establish an encrypted connection between the two services.

Links A list of links on which the Discovery Relay provides service. Each link identifier corresponds to a link identified by a previous 'Link Present' primitive sent by the Director, as described in Section 4.2.1.

#### 4.2.6. Discovery Relay Resigning

When a node providing Discovery Relay support can no longer continue to do so, it announces to the Director that it is no longer available using this primitive. The 'Discovery Relay Resigning' primitive includes the following data:

node identifier The node identifier of the Discovery Relay, unique among all nodes managed by a Director.

### 4.3. Orchestration System Behavior

#### 4.3.1. Link Present

The Director detects new links, or is configured with new links by the network operator. It is responsible for noticing that a link to which more than one participating node is connected is the same link. For example, see Section 6.1 of [RFC7788]. When a new link is detected, the Director reports the presence of that link to all enabled Discovery Proxy Performers, and to all Discovery Relay Performers. If the Director becomes aware of more than one link at the same time, or within an implementation-specific interval, it may announce the presence of more than one link at a time using the 'Link Present' primitive.

#### 4.3.2. Link Remove

The Director detects the removal of links, either as a result of routers that are connected to those links becoming unavailable, or as a result of manual changes to the configuration by the network operator. When a link that had previously been present is removed, the Director announces the removal of this link to all enabled Discovery Proxy performers and to all Discovery Relay performers. If the removal of more than one link is detected at the same time or within an implementation-specific interval, the removal of each such link may be announced in a single 'Link Remove' primitive.

#### 4.3.3. Discovery Proxy Available

When the Director receives a 'Discovery Proxy Available' primitive, it records the information in its list of available Discovery Proxies (henceforth "Discovery Proxy List"). If that node had previously reported that Discovery Proxy service was available, the entry in Discovery Proxy List for that node is replaced with an entry generated from the new update; any information in the previous entry that is not present in the update is discarded.

Whether or not the Director enables Discovery Proxy service on the Discovery Proxy announced in a newly-received 'Discovery Proxy Available' primitive is dependent on the operational model and configuration of that particular orchestration system, which is out of scope for this document. The same is true as to whether service discovery is enabled on all known links, or not. We assume here that Discovery Proxy service may be available but not enabled on some nodes, whereas Discovery Relay service is generally available, since it will only be used by enabled Discovery Proxies on interfaces on which service discovery is enabled.

If the Director enables Discovery Proxy service on that node, the Discovery Proxy is announced to all nodes currently providing Discovery Relay service, using 'Discovery Proxy Available' primitives. In addition, the set of all known Discovery Relays, and the information provided by them to the orchestration system, is announced to the node providing the Discovery Proxy service, using one or more 'Discovery Relay Available' primitives.

When a 'Discovery Proxy Available' primitive is received from a Discovery Proxy Performer for which service is already enabled, but the update includes different information than was present in the previous announcement, the Discovery Proxy service is re-announced to every Discovery Relay Performer.

#### 4.3.4. Discovery Proxy Resigning

When the Director receives a 'Discovery Proxy Resigning' primitive from a Discovery Proxy Performer that had previously sent a 'Discovery Proxy available' primitive, the Director first determines if Discovery Proxy service had been enabled on that node. If so, 'Discovery Proxy Resigning' notifications are sent to Discovery Relay Performers.

The Director may, as a result of a node's resignation from providing Discovery Proxy service, enable Discovery Proxy on some other node. If so, it does so as described in Section 4.3.3.

In addition to any announcements sent as a result of a node's resignation from providing Discovery Proxy service, the Director also looks for an entry in the Discovery Proxy List for that node. If one is present, it is removed.

#### 4.3.5. Discovery Relay Available

When the Director receives a 'Discovery Relay Available' primitive, it records the information in its list of available Discovery Relay Performers (henceforth "Discovery Relay List"). If that list already contains an entry for the Performer making the new report, the entry from the list is discarded and a new one generated from the new announcement.

Whether or not the Director enables service discovery through a particular Discovery Relay is dependent on the operation of that particular orchestration system, which is out of scope for this document. It is assumed that a Director may or may not enable a particular Discovery Relay.

If the Director enables service discovery through the relay that made the announcement, the relay is announced to all enabled Discovery Proxy Performers. In addition, if the relay had not previously been enabled for service discovery, the Director sends a 'Discovery Proxy Available' primitive to that Performer for each Discovery Proxy Performer on the Discovery Proxy List.

#### 4.3.6. Discovery Relay Resigning

When the Director receives a 'Discovery Relay Resigning' primitive, it checks to see if the node making the announcement had previously been listed as providing Discovery Relay service; if so, the entry for that node is removed from the list. If Discovery Relay service was enabled for that node, all nodes providing Discovery Proxy service are notified that this node is no longer providing Discovery Relay service, by sending a 'Discovery Relay Resigning' primitive to each such node.

#### 4.3.7. Node Available

The orchestration system may or may not track the coming and going of nodes that provide service discovery. If it does, depending on the operation of the system, it may be necessary to send some notification to the node to trigger its announcement of service discovery services. How this is done is out of scope for this document.

#### 4.3.8. Node Resigning

The orchestration system may or may not track the coming and going of nodes that provide service discovery. If it does, then when the departure of a node that has previously announced Discovery Relay and/or Discovery Proxy service should result in the synthesis of resignation events for those services on that node. The exact operation of this mechanism is out of scope for this document.

#### 4.4. Discovery Proxy Performer Behavior

Nodes may provide both Discovery Proxy and Discovery Relay service: the two services share no ports and are mutually compatible. When a node is providing both services, the behaviors described in this section are specific to the operation of the Discovery Proxy service on that node, not to the Discovery Relay service.



#### 4.4.1. Link Present

When a node that is providing Discovery Proxy service receives a link present notification, it checks to see if it currently has Discovery Relay service configured for each such link. For any such link for which it does not have Discovery Relay service configured, it identifies the set of Relay Proxies that provide service on that link. It then chooses a Discovery Relay node from this set using a random number generator. If it already has a connection to the Relay Proxy, it attempts to subscribe to mDNS messages from that link. If it does not have a connection, it attempts to establish one. If that succeeds, it attempts to subscribe to mDNS messages from that link. If the outcome of each of these attempts to get Discovery Relay service on the new link fails, it eliminates this Discovery Relay from the set and repeats the process until the set is empty.

If no attempt to subscribe to mDNS messages on the link is successful, then service discovery on that link is not possible. The Discovery Proxy node maintains a list of links on which Discovery Relay service is desired but not available; when an attempt to get Discovery Relay service on a link fails, either because no node is providing Discovery Relay service on that link, or because attempting to get service on that link from all nodes claiming to provide it has failed, the link is added to this list.

#### 4.4.2. Link Removed

When a link is removed, the Discovery Relay checks its list of connections to Discovery Relays for subscription for mDNS messages on that link. If one is present, the Discovery Relay unsubscribes from mDNS messages on that link. If there are no subscriptions present on that connection, the Discovery Relay terminates the connection. If the link is on the list of links for which Discovery Relay service is desired but not available, the link is removed from that list.

#### 4.4.3. Discovery Proxy Available

Discovery Proxy Performers send 'Discovery Proxy Available' primitives to the Director whenever their configuration changes in a way that affects the content of the primitive, and also whenever their node becomes newly available to the Director. In addition to notifying the Director when they first become connected to the Director's orchestration system, they must also notify the Director when they disconnect and reconnect.

When a node with Discovery Proxy service becomes available to the orchestration system, it informs the orchestration system that it can provide Discovery Proxy service. It also provides the orchestration

system with a list of IP addresses from which it may originate connections to Discovery Relays, and provides a TLS PKI cert or suitable bare public key which will be used for TLS Client Authentication.

Whenever the set of IP addresses from which the Discovery Proxy may initiate a connection to a Discovery Relay changes, the Discovery Proxy sends a new 'Discovery Proxy Available' primitive with its complete information, as above. It may be desirable for the Discovery Proxy node to choose a specific IP address from which all such connections will originate, so as to minimize the number of such updates that may be required, but this behavior is optional.

It is not ordinarily the case that the key or certificate used for authentication will change, but if it does, the Discovery Proxy node sends a complete new 'Discovery Proxy Available' primitive, which will contain the new key or certificate.

#### 4.4.4. Discovery Proxy Resigning

When a node that had previously provided Discovery Proxy service is no longer able to do so for any reason, it announces this to the orchestration system using a 'Discovery Proxy Resigning' primitive.

#### 4.4.5. Discovery Relay Available

When a node providing Discovery Proxy service receives a 'Discovery Relay Available' notification, it adds that Discovery Relay to its list of available Discovery Relays. If the Discovery Relay is already on the list, the information the list entry is compared to the new information provided in the 'Discovery Relay Available' primitive. If a connection to that Discovery Relay is present, and the destination IP address of that connection is no longer on the list of IP addresses supported by the Discovery Relay, or the public key of the Discovery Relay has changed, the connection is dropped and the process described in Section 4.4.6 is followed.

Otherwise, if there is a connection to the Discovery Relay, the list of links subscribed to on that connection is compared to the list of served links listed in the 'Discovery Relay Available' primitive; any links for which subscriptions exist that are not listed in the 'Discovery Relay Available' announcement are unsubscribed, and those links added to the list of links on which Discovery Relay service is not available.

At this point the process described in Section 4.4.1 is followed for each link on the list of links for which Discovery Relay service is not available.

#### 4.4.6. Discovery Relay Resigning

Discovery Relay drops its connection to that Discovery Relay and puts all links for which subscriptions existed on that connection onto the list of links on which Discovery Relay service is not available. Because it is possible that another Discovery Relay is available for that link, the Discovery Proxy node again follows the process described in Section 4.4.1.

#### 4.5. Discovery Relay Performer Behavior

Nodes that support service discovery may support both Discovery Proxy and Discovery Relay. Behaviors described here are specific to nodes that are providing Discovery Relay service. A node that provides both types of service will follow both the behavior described here and the behavior described for Discovery Proxy nodes.

##### 4.5.1. Link Present

When a Discovery Relay performer receives a link present notification, it determines for each link announced whether it has an interface that is directly connected to that link. If so, it determines whether it has previously announced the availability of service on that link. If not, it adds the link to the list of links on which it provides Discovery Relay service (henceforth "Discovery Relay link list").

If as a result of a 'Link Present' announcement the Discovery Relay link list has changed, the Discovery Relay performer sends a new 'Discovery Relay Available' primitive to the Director.

##### 4.5.2. Link Removed

When the Discovery Relay Performer receives a 'Link Removed' primitive, for each link mentioned in the primitive it checks to see if it is currently providing service on that link. For each link mentioned in the primitive for which it is providing service, it deletes that link from its list of links on which it is providing service. If any links were deleted from the list, the Discovery Relay Performer sends a new 'Discovery Relay Available' message to the Director.

##### 4.5.3. Discovery Proxy Available

Directors send 'Discovery Proxy Available' primitives to Discovery Relay Performers when new Discovery Proxy Performers announce their availability, and also when Discovery Proxy Performers announce changes to their configuration. When a Discovery Relay Performer

receives one of these primitives, it updates its Discovery Proxy IP address whitelist with the set of IP addresses from the primitive, and updates the Discovery Proxy authentication certificate as well. If the Discovery Proxy is connected to the Discovery Relay and either the certificate changed, or the source IP address of the connection is no longer on the whitelist, the Discovery Relay drops the connection.

#### 4.5.4. Discovery Proxy Resigning

Directors send 'Discovery Proxy Resigning' messages to Discovery Relay Performers when Discovery Proxy Performers indicate that they are no longer available, or when they are disabled by the orchestration system. When a Discovery Relay Performer receives this primitive, it checks to see if any connections from that Discovery Proxy are present. Any such connections are terminated.

#### 4.5.5. Discovery Relay Available

Discovery Relay Performers send 'Discovery Relay Available' primitives to the Director whenever their configuration changes in a way that affects the content of the primitive, and also whenever their node becomes newly available to the Director. In addition to notifying the Director when they first become connected to the Director's orchestration system, they must also notify the Director when they disconnect and reconnect.

Discovery Relays listen for connections from Discovery Proxies. Because no port is reserved for Discovery Relays, it is not useful to announce the availability of the service until the service is listening for connections, at which point it will know which port it is listening on. Therefore, before sending a 'Discovery Relay Available' primitive, a Discovery Relay Performer must have received its listening port from the Discovery Relay service.

#### 4.5.6. Discovery Relay Resigning

When a node providing Discovery Relay service must stop providing that service, it sends a 'Discovery Relay Resigning' primitive to the Director.

### 5. Connections between Discovery Proxies and Discovery Relays

When a Discovery Relay starts, it opens a passive TCP listener to receive connections from Discovery Proxies. This listener may be bound to one or more source IP addresses, or to the wildcard address, depending on the TCP implementation. When a connection is received, the relay must first validate that it is a connection to an IP

address to which connections are allowed. For example, it may be that only connections to ULAs are allowed, or to the IP addresses configured on certain interfaces. If the listener is bound to a specific IP address, this check is unnecessary.

The relay must then validate that the source IP address of the connection is on its whitelist. If the connection is not permitted either because of the source address or the destination address, the Discovery Relay responds to the TLS Client Hello message from the Discovery Proxy with a TLS user\_canceled alert ([I-D.ietf-tls-tls13] Section 6.1).

Otherwise, the Discovery Relay will attempt to complete a TLS handshake with the Discovery Proxy. Discovery Proxies are required to send the `post_handshake_auth` extension ([I-D.ietf-tls-tls13] Section 4.2.5). If a relay proxy receives a ClientHello message with no `post_handshake_auth` extension, the Discovery Relay rejects the connection with a `certificate_required` alert ([I-D.ietf-tls-tls13] Section 6.2).

Once the TLS handshake is complete, the Discovery Relay MUST request post-handshake authentication as described in ([I-D.ietf-tls-tls13] Section 4.6.2). If the Discovery Proxy refuses to send a certificate, or the key presented does not match the key associated with the IP address from which the connection originated, or the CertificateVerify does not validate, the connection is dropped with the TLS `access_denied` alert ([I-D.ietf-tls-tls13] Section 6.2).

Once the connection is established and authenticated, it is treated as a DNS TCP connection [RFC1035].

Aliveness of connections between Discovery Proxies and Relays is maintained as described in Section 4 of [I-D.ietf-dnsop-session-signal]. Discovery Proxies must also honor the 'Retry Delay' TLV (section 5 of [I-D.ietf-dnsop-session-signal]) if sent by the Discovery Relay.

Discovery Proxies may establish more than one connection to a specific Discovery Relay. This would happen in the case that a TCP connection stalls, and the Discovery Proxy is able to reconnect before the previous connection has timed out. It could also happen as a result of a server restart. It is not likely that two active connections from the same Discovery Proxy would be present at the same time, but it must be possible for additional connections to be established. The Discovery Relay may drop the old connection when the new one has been fully established, including a successful TLS handshake. What it means for two connections to be from the same Discovery Proxy is that the connections both have source addresses

that belong to the same proxy, and that they were authenticated using the same client certificate.

## 6. Traffic from Relays to Proxies

The mere act of connecting to a Discovery Relay does not result in any mDNS traffic being forwarded. In order to request that mDNS traffic from a particular link be forwarded on a particular connection, the Discovery Proxy must send a session signaling message containing one or more MDNS Link Request TLVs (Section 9.1) indicating the link from which traffic is requested.

When such a message is received, the Discovery Relay validates that each specified link is available for forwarding, and that forwarding is enabled for that link. For each such message the Discovery Relay validates each link specified and includes in a single response a list of zero or more MDNS Link Invalid TLVs (Section 9.2) for links that are not valid, and zero or more MDNS Link Subscribed TLVs (Section 9.3) for links that are valid. For each valid link, it begins forwarding all mDNS traffic from that link to the Discovery Proxy. Delivery is not guaranteed: if there is no buffer space, packets will be dropped. It is expected that regular mDNS retry processing will take care of retransmission of lost packets. The amount of buffer space is implementation dependent, but generally should not be more than the bandwidth delay product of the TCP connection [RFC1323].

mDNS messages from Relays to Proxies are framed within DNS Session Signaling messages. This allows multiple TLVs to be included. Each forwarded mDNS message is contained in an MDNS Message TLV (Section 9.4). The layer 2 source address of the message, if known, MAY be encoded in a Layer 2 Source TLV (Section 9.5). The source IP address of the message MUST be encoded in a IP Source Address TLV (Section 9.6). The source port of the message MUST be encoded in an IP Source port TLV (Section 9.7). The link on which the message was received MUST be encoded in a Link Identifier TLV (Section 9.8). The Discovery Proxy MUST silently ignore unrecognized TLVs in mDNS messages, and MUST NOT discard mDNS messages that include unrecognized TLVs.

A Discovery Proxy may discontinue listening for mDNS messages on a particular link by sending a session signaling message containing an MDNS Link Discontinue TLV (Section 9.9). Subsequent messages from that link that had previously been queued indicating may arrive. The Discovery Proxy should silently ignore such messages. The Discovery Relay MUST discontinue generating such messages as soon as the request is received. The Discovery Relay does not respond to this

message other than to discontinue forwarding mDNS messages from the specified links.

## 7. Traffic from Proxies to Relays

Like mDNS traffic from relays, each mDNS message sent by a Discovery Proxy to a Discovery Relay is encapsulated in an MDNS Message TLV (Section 9.4) within a session signaling message. Each message MUST contain one or more Link Identifier TLVs (Section 9.8). The Discovery Relay will transmit the message to the mDNS port and multicast address on each link. The message MUST include one or more IP family TLVs (Section 9.10). For each such TLVs that is included, the message will be sent on each link using the specified IP family. If no family codes are recognized, no packets will be transmitted.

## 8. Discovery Proxy Behavior

Discovery Proxies treat links for which Discovery Relay service is being used as if they were virtual interfaces; in other words, a Discovery Proxy serving multiple links using multiple Discovery Relays behaves the same as a Discovery Proxy serving multiple links using multiple physical network interfaces.

Discovery Proxies responding to mDNS messages for non-link-local IP addresses where the unicast bit is set respond directly, rather than through a proxy. Link-local responses are not supported for links to which Discovery Proxies are not directly connected.

## 9. Session Signaling TLVs

This document defines a modest number of new DNS Session Signaling TLVs.

### 9.1. MDNS Link Request

The MDNS Link Request TLV conveys a 32-bit link identifier from which a Discovery Proxy is requesting that a Discovery Relay forward mDNS traffic. The link identifier comes from the orchestration system (see Section 4.2.1). The SSOP-TYPE for this TLV is TBD1. The SSOP-LENGTH is always 4. The SSOP-DATA is the 32-bit identifier in network byte order.

### 9.2. MDNS Link Invalid

The MDNS Link Invalid TLV is returned in response to a session signaling message containing an MDNS Link Request, and returns the 32-bit identifier that was contained in that request. The link identifier comes from an MDNS Link Request TLV in the message being

responded to. The TLV indicates that the specified link identifier does not refer to a valid link, either because the link is not supported by the Discovery Relay, or because the identifier is not known. The SSOP-TYPE for this TLV is TBD2. The SSOP-LENGTH is always 4. The SSOP-DATA is the 32-bit identifier in network byte order.

### 9.3. MDNS Link Subscribed

The MDNS Link Subscribed TLV is returned in response to a session signaling message containing an MDNS Link Request, and returns the 32-bit identifier from a MDNS Link Request TLV that was contained in that request. It indicates that MDNS messages for the specified link have been successfully subscribed. The SSOP-TYPE for this TLV is TBD3. The SSOP-LENGTH is always 4. The SSOP-DATA is the 32-bit identifier in network byte order.

### 9.4. MDNS Message

The MDNS Message TLV is used to encapsulate an mDNS message that is being forwarded from a link to a Discovery Proxy, or is being forwarded from a Discovery Proxy to a link. The SSOP-TYPE for this TLV is TBD4. SSOP-LENGTH is the length of the application layer payload of the MDNS message. SSOP-DATA is the application layer payload of the message.

### 9.5. Layer 2 Source Address

The Layer 2 Source Address TLV is used to report the layer 2 address from which an mDNS message was received. This TLV is optionally present in session signaling messages from Discovery Relays to Discovery Proxies that contain mDNS messages when the source link-layer address is known. The SSOP-TYPE is TBD5. SSOP-LENGTH is variable, depending on the length of link-layer addresses on the link from which the message was received. SSOP-data is the link-layer address as it was received on the link.

### 9.6. IP Source Address

The IP Source Address TLV is used to report the IP source address from which an mDNS message was received. This TLV is present in session signaling messages from Discovery Relays to Discovery Proxies that contain mDNS messages. SSOP-TYPE is TBD6. SSOP-LENGTH is either 4, for an IPv4 address, or 16, for an IPv6 address. SSOP-DATA is the IP Address.



### 9.7. IP Source Port

The IP Source Port TLV is used to report the IP source port from which an mDNS message was received. This TLV is present in session signaling messages from Discovery Relays to Discovery Proxies. SSOP-TYPE is TBD7. SSOP-LENGTH is 2. SSOP-DATA is the source port in network byte order.

### 9.8. Link Identifier

This option is used both in session signaling messages from Discovery Proxies to Discovery Relays that contain mDNS messages, and in message from Discovery Relays to Discovery Proxies that contain mDNS messages. In the former case, it indicates a link to which the message should be forwarded; in the latter case, it indicates the link on which the message was received. SSOP-TYPE is TBD8. SSOP-LENGTH is 4. SSOP-DATA is a 32-bit link identifier as described in Section 4.2.1.

### 9.9. MDNS Discontinue

This option is used by Discovery Proxies to unsubscribe to mDNS messages on the specified link. More than one may be present in a single session signaling message. SSOP-TYPE is TBD9. SSOP-LENGTH is 4. SSOP-DATA is a 32-bit link identifier as described in Section 4.2.1.

### 9.10. IP Address Family

This option is used in mDNS messages sent by Discovery Proxies to links to indicate to the Discovery Relay which IP address family or families should be used when transmitting the message on the link. More than one may be present in a single session signaling message. SSOP-TYPE is TBD10. SSOP-LENGTH is 1. SSOP-DATA is a 8-bit IP family identifier. A value of 1 indicates IPv4. A value of 2 indicates IPv6. Other values are reserved, and MUST be ignored if not recognized.

## 10. Security Considerations

## 11. IANA Considerations

The IANA is kindly requested to update the DNS Session Signaling Type Codes Registry [I-D.ietf-dnsop-session-signal] by allocating codes for each of the TBD type codes listed in the following table, and by updating this document, here and in Section 9. Each type code should list this document as its reference document.

Opcode	Status	Name
TBD1	Standard	MDNS Link Request
TBD2	Standard	MDNS Link Invalid
TBD3	Standard	MDNS Link Subscribed
TBD4	Standard	MDNS Messsage
TBD5	Standard	Layer Two Source Address
TBD6	Standard	IP Source Address
TBD7	Standard	IP Destination Address
TBD8	Standard	Link Identifier
TBD9	Standard	MDNS Discontinue
TBD10	Standard	IP Address Family

DNS Session Signaling Type Codes to be allocated

## 12. IANA Considerations

## 13. Acknowledgments

## 14. References

### 14.1. Normative References

[I-D.ietf-dnsop-session-signal]

Bellis, R., Cheshire, S., Dickinson, J., Dickinson, S., Mankin, A., and T. Pusateri, "DNS Session Signaling", draft-ietf-dnsop-session-signal-02 (work in progress), March 2017.

[I-D.ietf-dnssd-hybrid]

Cheshire, S., "Discovery Proxy for Multicast DNS-Based Service Discovery", draft-ietf-dnssd-hybrid-06 (work in progress), March 2017.

[I-D.ietf-tls-tls13]

Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", draft-ietf-tls-tls13-20 (work in progress), April 2017.

[RFC1035]

Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <<http://www.rfc-editor.org/info/rfc1035>>.

[RFC1323]

Jacobson, V., Braden, R., and D. Borman, "TCP Extensions for High Performance", RFC 1323, DOI 10.17487/RFC1323, May 1992, <<http://www.rfc-editor.org/info/rfc1323>>.

- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<http://www.rfc-editor.org/info/rfc6241>>.
- [RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, DOI 10.17487/RFC6762, February 2013, <<http://www.rfc-editor.org/info/rfc6762>>.
- [RFC6763] Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", RFC 6763, DOI 10.17487/RFC6763, February 2013, <<http://www.rfc-editor.org/info/rfc6763>>.
- [RFC7788] Stenberg, M., Barth, S., and P. Pfister, "Home Networking Control Protocol", RFC 7788, DOI 10.17487/RFC7788, April 2016, <<http://www.rfc-editor.org/info/rfc7788>>.

#### 14.2. Informative References

- [TR-069] Broadband Forum, "CPE WAN Management Protocol", November 2013, <[https://www.broadband-forum.org/technical/download/TR-069\\_Amendment-5.pdf](https://www.broadband-forum.org/technical/download/TR-069_Amendment-5.pdf)>.

#### Authors' Addresses

Stuart Cheshire  
Apple Inc.  
1 Infinite Loop  
Cupertino, California 95014  
USA

Phone: +1 408 974 3207  
Email: [cheshire@apple.com](mailto:cheshire@apple.com)

Ted Lemon  
Nominum, Inc.  
800 Bridge Parkway  
Redwood City, California 94065  
United States of America

Phone: +1 650 381 6000  
Email: [ted.lemon@nominum.com](mailto:ted.lemon@nominum.com)

Internet Engineering Task Force  
Internet-Draft  
Intended status: Informational  
Expires: January 3, 2018

S. Cheshire  
Apple Inc.  
T. Lemon  
Nominum, Inc.  
July 2, 2017

Service Registration Protocol for DNS-Based Service Discovery  
draft-sctl-service-registration-00

Abstract

The DNS-SD Service Registration Protocol provides a way to perform DNS-Based Service Discovery using only unicast packets. This eliminates the dependency on Multicast DNS as the foundation layer, which has worked well in some environments, like the simplest of home networks, but not in others, like large enterprise networks (where multicast does not scale well to thousands of devices) and mesh networks (where multicast and broadcast are supported poorly, if at all). Broadly speaking, the DNS-SD Service Registration Protocol is DNS Update, with a few additions.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on January 3, 2018.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## 1. Introduction

DNS-Based Service Discovery [RFC6763] is a component of Zero Configuration Networking [RFC6760] [ZC] [Roadmap].

There are two facets of DNS-Based Service Discovery to consider: how relevant information makes its way into the DNS namespace (how a server offers its services to interested clients) and how clients access that information (how an interested client discovers and uses a service instance).

This document is concerned with the first of those two facets: how relevant information makes its way into the DNS namespace.

In the DNS-Based Service Discovery specification [RFC6763] Section 10 "Populating the DNS with Information" briefly discusses ways that relevant information can make its way into the DNS namespace. In the case of Multicast DNS [RFC6762], the relevant information trivially becomes visible in the ".local" namespace by virtue of devices answering for themselves. For unicast DNS names, ways that information makes its way into the DNS namespace include manual configuration of DNS zone files, possibly assisted using tools such as the "dns-sd -Z" command, automated tools such as a Discovery Proxy [DisProx], or explicit registration by the services themselves. It is the last option -- explicit registration by the services themselves -- that is the subject of this document.

## 2. Service Registration Protocol

The DNS-SD Service Registration Protocol is largely built on DNS Update [RFC2136] [RFC3007], with some additions.

When a device advertises services using Multicast DNS, the parent domain is implicitly ".local".

When a device advertises services in the traditional unicast DNS namespace, it needs to know the parent domain name for its services. This parent domain can be manually configured by a human operator, or learned from the network. In the DNS-SD specification [RFC6763] section 11, "Discovery of Browsing and Registration Domains (Domain Enumeration)", describes how a client device can learn a recommended default registration domain from the network.

In the remainder of this document, Section 3 covers cleanup of stale data, and Section 4 covers advertising services on behalf of devices that are sleeping to reduce power consumption.

The final question is security. Most dynamic DNS servers will not accept unauthenticated updates. In the case of manual configuration of registration domain by a human operator, the human operator can also configure an appropriate TSIG security key. In the case of automatic configuration via DNS-SD Domain Enumeration queries, it would be nice to also have zero-configuration security. While at first glance zero-configuration security may seem to be a self-contradiction, this document proposes a simple first-come first-served security mechanism, described below in Section 5.

### 3. Cleanup of Stale Data

The traditional DNS Update mechanisms [RFC2136] [RFC3007] implicitly assume they are being used by a human operator. If a human operator uses DNS Update (perhaps via the 'nsupdate' command) to create a record, then that record should stay created until the human operator decides to remove it.

The same assumptions do not apply to machine-generated records. If a mobile device creates one or more records using DNS Update, and later unceremoniously departs the network, then those stale records should eventually be removed.

The mechanism proposed here is modeled on DHCP. Just like a DHCP address lease, a record created using DNS Update has a lifetime. If the record is not refreshed before its lifetime expires, then the record is deleted.

When a client performs a DNS Update, it includes a EDNS(0) Update Lease option [DNS-UL]. The DNS Update Lease option indicates the requested lifetime of the records created or updated in the associated DNS Update message. In the DNS Update reply, the server returns its own EDNS(0) Update Lease option indicating the granted lifetime, which may be shorter, the same, or longer than the client requested. If the records are not refreshed before the granted lifetime expires, then the records are deleted.

DNS servers may be configured to refuse DNS Updates that do not include a DNS Update Lease option.

#### 4. Sleep Proxy

Another use of Service Registration Protocol is for devices that sleep to reduce power consumption.

In this case, in addition to the DNS Update Lease option [DNS-UL] described above, the device includes an EDNS(0) OWNER Option [Owner].

The DNS Update Lease option constitutes a promise by the device that it will wake up before this time elapses, to renew its records and thereby demonstrate that it is still attached to the network. If it fails to renew the records by this time, that indicates that it is no longer attached to the network, and its records should be deleted.

The EDNS(0) OWNER Option indicates that the device will be asleep, and will not be receptive to normal network traffic. When a DNS server receives a DNS Update with an EDNS(0) OWNER Option, that signifies that the DNS server should act as a proxy for any IPv4 or IPv6 address records in the DNS Update message. This means that the DNS server should send ARP or ND messages claiming ownership of the IPv4 and/or IPv6 addresses in the records in question. In addition, the DNS server should answer future ARP or ND requests for those IPv4 and/or IPv6 addresses, claiming ownership of them. When the DNS server receives a TCP SYN or UDP packet addressed to one of the IPv4 or IPv6 addresses for which it proxying, it should then wake up the sleeping device using the information in the EDNS(0) OWNER Option. At present version 0 of the OWNER Option specifies the "Wake-on-LAN Magic Packet" that needs to be sent; future versions could be extended to specify other wakeup mechanisms.



## 5. First-Come First-Served Naming

In some environments, such as home networks with an appropriate border gateway, it may be preferable to have some limited security on the protected internal network rather than no security at all.

Users have shown limited willingness to endure complicated configuration for their networked home devices. It is rare for home users to change even the factory-default name for their wireless printer, so it's questionable whether it's reasonable to expect them to configure passwords or security keys.

This document presents a zero-configuration first-come first-served naming mechanism.

Instead of requiring a preconfigured key installed by manual administration, a new device optimistically creates its DNS Service Discovery records, plus a DNS SIG(0) public key, using a DNS Update signed with its DNS SIG(0) private key.

The DNS server validates the signature on the message using the SIG(0) key already stored on the name, if present, and otherwise with the key sent in the update, if the requested name is not yet present. The server may check that the two public keys are the same before validating, and refuse the update if they are not, to avoid the cost of verifying the signature.

The lifetime of the DNS-SD PTR, SRV and TXT records [RFC6763] is typically set to two hours. That way, if a device is disconnected from the network, its stale data does not persist for too long, advertising a service that is not accessible.

However, the lifetime of its DNS SIG(0) public key should be set to a much longer time, typically 14 days. The result of this is that even though a device may be temporarily unplugged, disappearing from the network for a few days, it makes a claim on its name that lasts much longer.

This way, even if a device is unplugged from the network for a few days, and its services are not available for that time, no other rogue device can come along and immediately claim its name the moment it disappears from the network. It takes a much longer time before an abandoned name becomes available for re-use.

When using this first-come first-served security mechanism, the server accepting or rejecting the updates utilizes knowledge of the DNS-Based Service Discovery semantics [RFC6763]. Specifically, for all records aside from PTR records, the update must be validly signed

using the SIG(0) key with the same DNS resource record owner name (the name on the left in a traditional textual zone file). For additions or deletions of PTR records, the update must be validly signed using the SIG(0) key with the same DNS resource record owner name as the rdata in the PTR record (the name on the right in a traditional textual zone file).

## 6. Security Considerations

To be completed.

## 7. References

### 7.1. Normative References

- [RFC6763] Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", RFC 6763, DOI 10.17487/RFC6763, February 2013, <<http://www.rfc-editor.org/info/rfc6763>>.

### 7.2. Informative References

- [RFC2136] Vixie, P., Ed., Thomson, S., Rekhter, Y., and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)", RFC 2136, DOI 10.17487/RFC2136, April 1997, <<http://www.rfc-editor.org/info/rfc2136>>.
- [RFC3007] Wellington, B., "Secure Domain Name System (DNS) Dynamic Update", RFC 3007, DOI 10.17487/RFC3007, November 2000, <<http://www.rfc-editor.org/info/rfc3007>>.
- [RFC6760] Cheshire, S. and M. Krochmal, "Requirements for a Protocol to Replace the AppleTalk Name Binding Protocol (NBP)", RFC 6760, DOI 10.17487/RFC6760, February 2013, <<http://www.rfc-editor.org/info/rfc6760>>.
- [RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, DOI 10.17487/RFC6762, February 2013, <<http://www.rfc-editor.org/info/rfc6762>>.
- [DisProx] Cheshire, S., "Discovery Proxy for Multicast DNS-Based Service Discovery", draft-ietf-dnssd-hybrid-06 (work in progress), March 2017.
- [DNS-UL] Sekar, K., "Dynamic DNS Update Leases", draft-sekar-dns-ul-01 (work in progress), August 2006.
- [Roadmap] Cheshire, S., "Service Discovery Road Map", draft-cheshire-dnssd-roadmap-00 (work in progress), July 2017.
- [Owner] Cheshire, S. and M. Krochmal, "EDNS0 OWNER Option", draft-cheshire-edns0-owner-option-01 (work in progress), July 2017.
- [ZC] Cheshire, S. and D. Steinberg, "Zero Configuration Networking: The Definitive Guide", O'Reilly Media, Inc. , ISBN 0-596-10100-7, December 2005.

Authors' Addresses

Stuart Cheshire  
Apple Inc.  
1 Infinite Loop  
Cupertino, California 95014  
USA

Phone: +1 408 974 3207  
Email: cheshire@apple.com

Ted Lemon  
Nominum, Inc.  
800 Bridge Parkway  
Redwood City, California 94065  
United States of America

Phone: +1 650 381 6000  
Email: ted.lemon@nominum.com

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: September 14, 2017

T. Lemon  
Nominum, Inc.  
D. Migault  
Ericsson  
March 13, 2017

Simple Homenet Naming and Service Discovery Architecture  
draft-tldm-simple-homenet-naming-00

Abstract

This document describes a simple name resolution and service discovery architecture for homenets. This architecture covers local publication of names, as well as name resolution for local and global names.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on September 14, 2017.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
1.1. Existing solutions . . . . .	3
2. Terminology . . . . .	4
3. Name Resolution . . . . .	4
3.1. Configuring Resolvers . . . . .	4
3.2. Configuring Service Discovery . . . . .	5
3.3. Resolution of local names . . . . .	5
3.4. DNSSEC Validation . . . . .	6
3.5. Support for Multiple Provisioning Domains . . . . .	6
3.6. Using the Local Namespace While Away From Home . . . . .	7
4. Management Considerations . . . . .	7
5. Privacy Considerations . . . . .	7
6. Security Considerations . . . . .	8
7. IANA considerations . . . . .	8
8. Normative References . . . . .	8
Authors' Addresses . . . . .	9

## 1. Introduction

Associating domain names with hosts on the Internet is a key factor in enabling communication with hosts, particularly for service discovery. This document describes a simple way of providing name service and service discovery for homenets. In principle, it may make sense to be able to publish names of devices on the homenet, so that services on the homenet can be accessed outside of the homenet. Such publication is out of scope for this document. It may be desirable to secure the homenet zone using DNSSEC. This is likewise out of scope for this document.

In order to provide name service, several provisioning mechanisms must be available:

- o Provisioning of a domain name under which names can be published and services advertised
- o Associating names that are subdomains of that name with hosts.
- o Advertising services available on the local network by publishing resource records on those names.
- o Distribution of names published in that namespace to servers that can be queried in order to resolve names
- o Correct advertisement of name servers that can be queried in order to resolve names

- o Timely removal of published names and resource records when they are no longer in use

Homenet adds the following considerations:

1. Some names may be published in a broader scope than others. For example, it may be desirable to advertise some homenet services to users who are not connected to the homenet. However, it is unlikely that all services published on the home network would be appropriate to publish outside of the home network. In many cases, no services will be appropriate to publish outside of the network, but the ability to do so is required.
2. Users cannot be assumed to be skilled or knowledgeable in name service operation, or even to have any sort of mental model of how these functions work. All of the operations mentioned here must reliably function automatically, without any user intervention or debugging.
3. Because user intervention cannot be required, naming conflicts must be resolved automatically, and, to the extent possible, transparently.
4. Hosts that do not implement any homenet-specific capabilities must still be able to discover and access services on the homenet, to the extent possible.
5. Devices that provide services must be able to publish those services on the homenet, and those services must be available from any part of the homenet, not just the link to which the device is attached.
6. Homenet explicitly supports multihoming--connecting to more than one Internet Service Provider--and therefore support for multiple provisioning domains [6] is required to deal with situations where the DNS may give a different answer depending on whether caching resolvers at one ISP or another are queried.

#### 1.1. Existing solutions

Previous attempts to automate naming and service discovery in the context of a home network are able to function with varying degrees of success depending on the topology of the home network. For example, Multicast DNS [4] can provide naming and service discovery [5], but only within a single multicast domain.

The Domain Name System provides a hierarchical namespace [1], a mechanism for querying name servers to resolve names [2], a mechanism

for updating namespaces by adding and removing names [3], and a mechanism for discovering services [5]. Unfortunately, DNS provides no mechanism for automatically provisioning new namespaces, and secure updates to namespaces require pre-shared keys, which won't work for an unmanaged network. DHCP can be used to populate names in a DNS namespace; however at present DHCP cannot provision service discovery information.

Hybrid Multicast DNS [7] proposes a mechanism for extending multicast DNS beyond a single multicast domain.. However, it has serious shortcomings as a solution to the Homenet naming problem. The most obvious shortcoming is that it requires that every multicast domain have a separate name. This then requires that the homenet generate names for every multicast domain, and requires that the end user have a mental model of the topology of the network in order to guess on which link a given service may appear. [xxx is this really true at the UI?]

## 2. Terminology

This document uses the following terms and abbreviations:

HNR Homenet Router

ISP Internet Service Provider

GNRP Global Name Registration Provider

## 3. Name Resolution

### 3.1. Configuring Resolvers

Hosts on the homenet receive a set of resolver IP addresses using either DHCP or RA. IPv4-only hosts will receive IPv4 addresses of resolvers, if available, over DHCP. IPv6-only hosts will receive resolver IPv6 addresses using either stateful (if available) or stateless DHCPv6, or through the domain name option in router advertisements. All homenet routers provide resolver information using both stateless DHCPv6 and RA; support for stateful DHCPv6 and DHCPv4 is optional, however if either service is offered, resolver addresses will be provided using that mechanism as well. Resolver IP addresses will always be IP addresses on the local link: every HNR is required to provide name resolution service. This is necessary to allow DNS update using presence on-link as a mechanism for rejecting off-network attacks.



### 3.2. Configuring Service Discovery

DNS-SD uses several default domains for advertising local zones that are available for service discovery. These include the `'local'` domain, which is searched using mDNS, and also the IPv4 and IPv6 reverse zone corresponding to the prefixes in use on the local network. For the homenet, no support for queries against the `"local"` zone is provided by HNRs: a `"local"` query will be satisfied or not by services present on the local link. This should not be an issue: all known implementations of DNS-SD will do unicast queries using the DNS protocol.

Service discovery is configured using the technique described in Section 11 of DNS-Based Service Discovery [5]. HNRs will answer domain enumeration queries against every IPv4 address prefix advertised on a homenet link, and every IPv6 address prefix advertised on a homenet link, including prefixes derived from the homenet's ULA(s). Whenever the `"<domain>"` sequence appears in this section, it references each of the domains mentioned in this paragraph.

Homenets advertise the availability of several browsing zones in the `"b._dns_sd.<domain>"` subdomain. By default, the `TBD1` domain is advertised. Similarly, `TBD1` is advertised as the default browsing and service registration domain under `"db._dns_sd.<domain>"`, `"r._dns_sd.<domain>"`, `"dr._dns_sd.<domain>"` and `"lb._dns_sd.<domain>"`.

### 3.3. Resolution of local names

Local names appear as subdomains of `[TBD1]`. These names can only be resolved within the homenet; not only is `[TBD1]` not a globally unique name, but queries from outside of the homenet for any name, on or off the homenet, must be rejected with a `REFUSED` response.

In addition, names can appear as subdomains of the locally-served `'in-addr.arpa'` or `'ip6.addr'` zone that corresponding to the ULA that is in use on the homenet. IP addresses and names advertised locally MUST use the homenet's ULA.

It is possible that local services may number themselves using more than one of the prefixes advertised locally. Homenet hybrid proxies MUST filter out global IP addresses, providing only ULA addresses, similar to the process described in section 5.5.2 of [7]. [xxx is this going to be a problem?]

The Hybrid Proxy model relies on each link having its own name. However, homenets do not actually have a way to name local links that

will make any sense to the end user. Consequently, this mechanism will not work. In order to paper over this, some changes are required:

- o The Hybrid Proxy function is divided into two: relaying proxies, and aggregating proxies. There must be exactly one querying proxy per link; there can be as few as one aggregating proxy per homenet.
- o Relaying proxies do no translation, for example from ".local" to "bldg1.example.com" as shown in section 5.3 of [7]. They simply take queries over the DNS protocol for names in subdomains of '.local', the link-specific 'ip6.addr', and the link-specific 'in-addr.arpa' zones, and respond with the exact answers received.
- o There must be exactly one querying proxy per internal link on the homenet; for links that are connected to more than one homenet router, HNCP is used to choose which router will provide the service.
- o Querying proxies perform translation. Machine readable names are presented as subdomains of the TBD1 domain. Human readable names are presented as subdomains of the \_hr.TBD1 domain.
- o Every homenet router can provide a querying proxy, or only one router can. This is determined by HNCP; all homenet routers must provide this capability, but some homenet routers may provide enhanced querying proxy capabilities such that homenet routers providing only those capabilities described in this document must be disabled. Therefore, all homenet routers must be able to act as a querying proxy, or forward DNS queries to a central querying proxy, according to what is specified through HNCP.

### 3.4. DNSSEC Validation

DNSSEC Validation for the TBD1 zone and for the locally-served 'ip6.arpa' and 'in-addr.arpa' domains is not possible without a trust anchor. Establishment of a trust anchor for such validation is out of scope for this document.

### 3.5. Support for Multiple Provisioning Domains

Homenets must support the Multiple Provisioning Domain Architecture [6]. In order to support this architecture, each homenet router that provides name resolution must provide one resolver for each provisioning domain (PvD). Each homenet router will advertise one resolver IP address for each PvD. DNS requests to the resolver associated with a particular PvD, e.g. using RA options [8] will be

resolved using the external resolver(s) provisioned by the service provider responsible for that PvD.

The homenet is a separate provisioning domain from any of the service providers. The global name of the homenet can be used as a provisioning domain identifier, if one is configured. Homenets should allow the name of the local provisioning domain to be configured; otherwise by default it should be "Home Network xxx", where xxx is the generated portion of the homenet's ULA prefix, represented as a base64 string.

The resolver for the homenet PvD is offered as the primary resolver in RAs and through DHCPv4 and DHCPv6. When queries are made to the homenet-PvD-specific resolver for names that are not local to the homenet, the resolver will use a round-robin technique, alternating between service providers with each step in the round-robin process, and then also between external resolvers at a particular service provider if a service provider provides more than one. The round-robinning should be done in such a way that no service provider is preferred, so if service provider A provides one caching resolver (A), and service provider B provides two (B1, B2), the round robin order will be (A, B1, A, B2), not (A, B1, B2).

Every resolver provided by the homenet, regardless of which provisioning domain it is intended to serve, will accept updates for subdomains of the TBD1 and locally-served 'ip6.arpa' and 'in-addr.arpa' domains from hosts on the local link.

### 3.6. Using the Local Namespace While Away From Home

This architecture does not provide a way for service discovery to be performed on the homenet by devices that are not directly connected to a link that is part of the homenet.

## 4. Management Considerations

This architecture is intended to be self-healing, and should not require management. That said, a great deal of debugging and management can be done simply using the DNS service discovery protocol.

## 5. Privacy Considerations

Privacy is somewhat protected in the sense that names published on the homenet are only visible to devices connected to the homenet. This may be insufficient privacy in some cases.

The privacy of host information on the local net is left to hosts. Various mechanisms are available to hosts to ensure that tracking does not occur if it is not desired. However, devices that need to have special permission to manage the homenet will inevitably reveal something about themselves when doing so. It may be possible to use something like HTTP token binding[9] to mitigate this risk.

## 6. Security Considerations

There are some clear issues with the security model described in this document, which will be documented in a future version of this section. A full analysis of the avenues of attack for the security model presented here have not yet been done, and must be done before the document is published.

## 7. IANA considerations

This document is relying on the allocation of [TBD1] described in Special Use Top Level Domain '.homenet' [10]. As such, no new actions are required by IANA, but this document can't proceed until that allocation is done. At that time, the name [TBD1] can be substituted for the name that is eventually allocated during the processing of that document.

## 8. Normative References

- [1] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <<http://www.rfc-editor.org/info/rfc1034>>.
- [2] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <<http://www.rfc-editor.org/info/rfc1035>>.
- [3] Vixie, P., Ed., Thomson, S., Rekhter, Y., and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)", RFC 2136, DOI 10.17487/RFC2136, April 1997, <<http://www.rfc-editor.org/info/rfc2136>>.
- [4] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, DOI 10.17487/RFC6762, February 2013, <<http://www.rfc-editor.org/info/rfc6762>>.
- [5] Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", RFC 6763, DOI 10.17487/RFC6763, February 2013, <<http://www.rfc-editor.org/info/rfc6763>>.

- [6] Anipko, D., Ed., "Multiple Provisioning Domain Architecture", RFC 7556, DOI 10.17487/RFC7556, June 2015, <<http://www.rfc-editor.org/info/rfc7556>>.
- [7] Cheshire, S., "Hybrid Unicast/Multicast DNS-Based Service Discovery", draft-ietf-dnssd-hybrid-05 (work in progress), November 2016.
- [8] Korhonen, J., Krishnan, S., and S. Gundavelli, "Support for multiple provisioning domains in IPv6 Neighbor Discovery Protocol", draft-ietf-mif-mpvd-ndp-support-03 (work in progress), February 2016.
- [9] Popov, A., Nystrom, M., Balfanz, D., Langley, A., and J. Hodges, "Token Binding over HTTP", draft-ietf-tokbind-https-08 (work in progress), February 2017.
- [10] Pfister, P. and T. Lemon, "Special Use Top Level Domain '.homenet'", draft-ietf-homenet-dot-03 (work in progress), March 2017.

#### Authors' Addresses

Ted Lemon  
Nominum, Inc.  
800 Bridge Parkway  
Redwood City, California 94065  
United States of America

Phone: +1 650 381 6000  
Email: [ted.lemon@nominum.com](mailto:ted.lemon@nominum.com)

Daniel Migault  
Ericsson  
8400 boulevard Decarie  
Montreal, QC H4P 2N2  
Canada

Email: [daniel.migault@ericsson.com](mailto:daniel.migault@ericsson.com)