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

S. Cheshire
Apple Inc.

July 8, 2016

Special Use Top Level Domain 'home'
draft-cheshire-homenet-dot-home-03

Abstract

This document specifies usage of the top-level domain "home", for names that are meaningful and resolvable within some scope smaller than the entire global Internet, but larger than the single link supported by Multicast DNS.

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

1. Introduction

Globally unique domain names are available to individuals and organizations for a modest annual fee. However, there are situations where a globally unique domain name is not available, or has not yet been configured, and in these situations it is still desirable to be able to use DNS host names [RFC1034] [RFC1035], DNS-Based Service Discovery [RFC6763], and other facilites built on top of DNS.

In the absence of available globally unique domain names, Multicast DNS [RFC6762] makes it possible to use DNS-like facilities with names that are unique within the local link, using the "local" top-level domain.

This document specifies usage of a similar top-level domain, "home", for names that have scope larger than a single link, but smaller than the entire global Internet.

Evidence indicates that ".home" queries frequently leak out and reach the root name servers [ICANN1][ICANN2]. This appears to be because of widespread usage of ".home" names in home networks, for example to name a printer "printer.home." When a user takes their laptop to a public Wi-Fi hotspot, attempts by that laptop to contact that printer result in fruitless ".home" queries to the root name servers. It would be beneficial for operators of public Wi-Fi hotspots to recognize and (negatively) answer such queries locally, thereby reducing unnecessary load on the root name servers, and this document would give those operators the authority to do that. Readers who are aware of other usages of ".home" names, that are not compatible with the rules proposed here, are encouraged to contact the authors with information to help revise and improve this draft.

It is expected that the rules for ".home" names outlined here will also be suitable to meet the needs of the IETF HOMENET Working Group, though that is not the primary goal of this document. The primary goal of this draft is to understand and document the current usage. If the needs of the IETF HOMENET Working Group are not met by this document codifying the current de facto usage, then the Working Group may choose to reserve a different Special Use Domain Name [RFC6761] which does meet their needs. With luck that may not be necessary, and a single document may turn out to be sufficient to serve both purposes. In any case, the HOMENET Working Group is likely to be a good community in which to find knowledge about how ".home" names are currently used.

[Author's Note, to be removed when document is published: The purpose of this draft is not to propose some novel new usage for ".home" names. The purpose is to document and describe the observed current widespread behavior, and to solicit feedback about whether that description is accurate.]

2. Conventions and Terminology Used in this Document

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 "Key words for use in RFCs to Indicate Requirement Levels" [RFC2119].

3. Mechanism

Typical residential home gateways configure their local clients via DHCP [RFC2131]. In addition to the client's IP address, this DHCP configuration information typically also includes other configuration parameters, like the IP address of the recursive (caching) DNS server the client is to use, which is usually the home gateway's own address (the home gateway is also a DNS cache/relay).

For a home network consisting of just a single link (or several physical links bridged together to appear as a single logical link to IP) Multicast DNS [RFC6762], which requires no configuration, is sufficient for client devices to look up the dot-local host names of peers on the same home network, and perform DNS-Based Service Discovery (DNS-SD) [RFC6763] of services offered on that home network.

For a home network consisting of multiple links that are interconnected using IP-layer routing instead of link-layer bridging, link-local Multicast DNS alone is insufficient because link-local Multicast DNS packets, by design, do not cross between links. (This was a deliberate design choice for Multicast DNS, since even on a single link multicast traffic is expensive -- especially on Wi-Fi links -- and multiplying the amount of multicast traffic by flooding it across multiple links would make that problem even worse.) In this environment, Unicast DNS packets (as may be facilitated by use of ".home" names instead of ".local" names) should be used for cross-link name resolution and service discovery.

For residential home networks, Zero Configuration [ZC] operation is desirable, without requiring any manual configuration from the user. A client device learns about its network environment in a variety of ways. It builds a list of network-recommended DNS search domains using DHCP options 15 (Domain Name option [RFC2132]) and 119 (Domain Search option [RFC3397]). It builds a list of network-recommended DNS-SD browsing domains by sending domain enumeration queries [RFC6763].

For organizations and individuals with a registered globally unique domain name under their control, hosts and services can be given names within that domain. Client devices can be configured to use that globally unique domain name as their DNS search domain and/or DNS-SD browsing domain [RFC6763]. For example, at IETF meetings the network configures client devices to use "meeting.ietf.org." as their DNS search domain and DNS-SD browsing domain. This domain name is globally unique and under the control of the IETF. It is entered into the DHCP and DNS servers manually by the IETF meeting network administrators, and then communicated automatically via the network

to client devices.

When a suitable globally unique domain name is available, as at IETF meetings, manual configuration of that name in a residential home gateway (or equivalent enterprise equipment) is appropriate. The network infrastructure then communicates that information to clients, without any additional manual configuration required on those clients.

However, many residential customers do not have any registered globally unique domain name available. This may be because they don’t want to pay the annual fee, or because they are unaware of the process for obtaining one, or because they are simply uninterested in having their own globally unique domain. This category also includes customers who intend to obtain a globally unique domain, but have not yet done so. For these users, it would be valuable to be able to perform cross-link name resolution and service discovery using Unicast DNS without requiring a globally unique domain name.

To facilitate zero configuration operation, residential home gateways should be sold preconfigured with the default unicast domain name "home". This default unicast domain name is not globally unique, since many different residential home gateways will be using the name "home" at the same time, but is sufficient for useful operation within a small collection of links. Such residential home gateways SHOULD offer a configuration option to allow the default (non-unique) unicast domain name to be replaced with a globally unique domain name for cases where the customer has a globally unique domain available and wishes to use it.

This use of the the top-level domain "home" for private local use is not new. Many home gateways have been using the name this way for many years, and it remains in widespread use, as evidenced by the large volume of invalid queries for "home" reaching the root name servers [ICANN1][ICANN2]. The current root server traffic load is due to things like home gateways configuring clients with "home" as a search domain, and then leaking the resulting dot-home queries upstream. In large part what the document proposes is, "stop leaking dot-home queries upstream." This document codifies the existing practice, and provides formal grounds basis for ISPs to legitimately block such queries in order to reduce unnecessary load on the root name servers.

4. Security Considerations

Users should be aware that, like private IP addresses [RFC1918], names in the "home" domain have only local significance. The name "My-Printer.home" in one location may not reference the same device as "My-Printer.home" in a different location.

5. IANA Considerations

[Once published, this should say] IANA has recorded the top-level domain "home" in the Special-Use Domain Names registry [SUDN].

5.1. Domain Name Reservation Considerations

The top-level domain "home", and any names falling within that domain (e.g., "My-Computer.home.", "My-Printer.home.", "_ipp._tcp.home."), are special [RFC6761] in the following ways:

1. Users may use these names as they would other DNS names, entering them anywhere that they would otherwise enter a conventional DNS name, or a dotted decimal IPv4 address, or a literal IPv6 address.

   Since there is no global authority responsible for assigning dot-home names, devices on different parts of the Internet could be using the same name. Users SHOULD be aware that using a name like "www.home" may not actually connect them to the web site they expected, and could easily connect them to a different web page, or even a fake or spoof of their intended web site, designed to trick them into revealing confidential information. As always with networking, end-to-end cryptographic security can be a useful tool. For example, when connecting with ssh, the ssh host key verification process will inform the user if it detects that the identity of the entity they are communicating with has changed since the last time they connected to that name.

2. Application software may use these names the same way it uses traditional globally unique Unicast DNS names, and does not need to recognize these names and treat them specially in order to work correctly. This document specifies the use of the top-level domain "home" in on-the-wire messages. Ideally this would be purely a protocol-level identifier, not seen by end users. However, in some applications domain names are seen by end users, and in those cases, the protocol-level identifier "home" becomes visible, even for users for whom English is not their preferred language. For this reason, applications MAY choose to use additional UI cues (icon, text color, font, highlighting, etc.)

to communicate to the user that this is a special name with special properties. Due to the relative ease of spoofing dot-home names, end-to-end cryptographic security remains important when communicating across a local network, just as it is when communicating across the global Internet.

3. Name resolution APIs and libraries SHOULD NOT recognize these names as special and SHOULD NOT treat them differently. Name resolution APIs SHOULD send queries for these names to their configured recursive/caching DNS server(s).

4. Recursive/caching DNS servers SHOULD recognize these names as special and SHOULD NOT, by default, attempt to look up NS records for them, or otherwise query authoritative DNS servers in an attempt to resolve these names. Instead, recursive/caching DNS servers SHOULD, by default, act as authoritative and generate immediate responses for all such queries. This is to avoid unnecessary load on the root name servers and other name servers.

   The type of response generated depends on the role of the recursive/caching DNS server: (i) Traditional recursive DNS servers (such as those run by ISPs providing service to their customers) SHOULD, by default, generate immediate negative responses for all such queries. (ii) Recursive/caching DNS servers incorporated into residential home gateways of the kind described by this document should act as authoritative for these names and return positive or negative responses as appropriate.

   Recursive/caching DNS servers MAY offer a configuration option to enable upstream resolving of these names, for use in networks where these names are known to be handled by an authoritative DNS server in said private network. This option SHOULD be disabled by default, and SHOULD be enabled only when appropriate, to avoid queries leaking out of the private network and placing unnecessary load on the root name servers.

5. Traditional authoritative DNS servers SHOULD recognize these names as special and SHOULD, by default, generate immediate negative responses for all such queries, unless explicitly configured otherwise by the administrator. As described above, DNS servers incorporated into residential home gateways of the kind described by this document should act as authoritative for these names and return positive or negative responses as appropriate, unless explicitly configured otherwise by the administrator.

6. DNS server operators SHOULD, if they are using these names, configure their authoritative DNS servers to act as authoritative

for these names. In the case of zero-configuration residential home gateways of the kind described by this document, this configuration is implicit in the design of the product, rather than a result of conscious administration by the customer.

7. DNS Registries/Registrars MUST NOT grant requests to register these names in the normal way to any person or entity. These names are reserved for use in private networks and fall outside the set of names available for allocation by registries/ registrars. Attempting to allocate a these name as if it were a normal DNS domain name will probably not work as desired, for reasons 4, 5, and 6 above.

6. Acknowledgments

Thanks to Francisco Arias of ICANN for his review and comments on this draft.

7. References

7.1. 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>.

[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>.

[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>.

7.2. Informative References

[RFC1918] Rekhter, Y., Moskowitz, B., Karrenberg, D., J. 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>.

[RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, DOI 10.17487/RFC2131, March 1997, <http://www.rfc-editor.org/info/rfc2131>.

[RFC2132] Alexander, S. and R. Droms, "DHCP Options and BOOTP Vendor Extensions", RFC 2132, DOI 10.17487/RFC2132, March 1997, <http://www.rfc-editor.org/info/rfc2132>.

[RFC3397] Aboba, B. and S. Cheshire, "Dynamic Host Configuration Protocol (DHCP) Domain Search Option", RFC 3397, DOI 10.17487/RFC3397, November 2002, <http://www.rfc-editor.org/info/rfc3397>.

[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>.

[ICANN1] "New gTLD Collision Risk Mitigation", <https://www.icann.org/en/about/staff/security/ssr/new-gtld-collision-mitigation-05aug13-en.pdf>.

[ICANN2] "New gTLD Collision Occurrence Management", <https://www.icann.org/en/system/files/files/resolutions-new-gtld-annex-1-07oct13-en.pdf>.

[SUDN] "Special-Use Domain Names Registry", <http://www.iana.org/assignments/special-use-domain-names/>.

[ZC] Cheshire, S. and D. Steinberg, "Zero Configuration Networking: The Definitive Guide", O'Reilly Media, Inc. , ISBN 0-596-10100-7, December 2005.

Author's Address

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

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

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

J. Chroboczek
IRIF, University of Paris-Diderot

July 8, 2016

Homenet profile of the Babel routing protocol
draft-ietf-homenet-babel-profile-00

Abstract

This document defines the subset of the Babel routing protocol [RFC6126] and its extensions that a Homenet router must implement.

Table of Contents

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 defines the exact subset of the Babel protocol and its extensions that is required by a conformant implementation of the Homenet protocol suite.

1.1. Background

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

- 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.

- 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.

- 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].

- The extension mechanism for Babel is defined in RFC 7557 [RFC7557].

- 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

[Sentences within square brackets are editorial notes and are not intended for publication.]

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.

[BS suggest that this should be a MUST.]

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 implement HMAC-based authentication, as defined in RFC 7298, MUST implement the two mandatory-to-implement algorithms defined in RFC 7298, and MUST enable and require authentication when instructed to do so by HNCP.

Rationale: some home networks include "guest" links that can be used by third parties that are not necessarily fully trusted. In such networks, it is essential that either the routing protocol is secured or the guest links are carefully firewalled.

Generic mechanisms such as DTLS and dynamically keyed IPsec are not able to protect multicast traffic, and are therefore difficult to use with Babel. Statically keyed IPsec, perhaps with keys rotated by HNCP, is vulnerable to replay attacks and would therefore require the addition of a nonce mechanism to Babel.

[There is no consensus about this requirement. A simpler solution is to disable Babel on guest interfaces. MS suggests this might be a SHOULD.]

[This needs expanding with an explanation of how HNCP is supposed to signal the use of authentication.]

REQ6: 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 lead to sub-optimal routing.

REQ7: 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".

[This should probably be MUST, but it might be difficult or even impossible to implement in some environments, especially in the presence of wired-to-wireless bridges.]

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: delay-based routing is useful in redundant meshes of tunnels, which do not occur in typical home networks (which typically use at most one VPN link). Interference-aware routing, on the other hand, is likely to be useful in home networks, but the extension requires further evaluation before it can be recommended for widespread deployment.

3. Acknowledgments

4. References

4.1. 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.

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

[RFC7557] Chroboczek, J., "Extension Mechanism for the Babel Routing Protocol", RFC 7557, May 2015.

4.2. 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

[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>.

Author’s Address

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

Email: jch@pps.univ-paris-diderot.fr

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 19, 2019

J. Chroboczek
IRIF, University of Paris-Diderot

July 18, 2018

Homenet profile of the Babel routing protocol
draft-ietf-homenet-babel-profile-07

Abstract

This document defines the exact subset of the Babel routing protocol and its extensions that is required by an implementation of the Homenet protocol suite, as well as the interactions between the Home Networking Control Protocol (HNCP) and Babel.

Table of Contents

1. Introduction

The core of the Homenet protocol suite consists of the Home Networking Control Protocol (HNCP) [RFC7788], a protocol used for flooding configuration information and assigning prefixes to links, combined with the Babel routing protocol [RFC6126bis]. Babel is an extensible, flexible and modular protocol: minimal implementations of Babel have been demonstrated that consist of a few hundred 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. Requirement 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.

1.2. Background

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

- RFC 6126bis [RFC6126bis] defines the Babel routing protocol. It allows Babel's control data to be carried either over link-local IPv6 or over 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 RFC 6126bis Babel will interoperate, in the sense that they will maintain a set of loop-free forwarding paths. However, if they implement conflicting options, they might not be able to exchange a full set of routes; in the worst case, an implementation that only implements the IPv6 subset of the protocol and an implementation that only implements the IPv4 subset of the protocol will not exchange any routes. In addition, if implementations use conflicting route selection policies, persistent oscillations might occur.

- The informative Appendix A of RFC 6126bis 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.

- While RFC 6126bis 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].

- Five RFCs and Internet-Drafts define optional extensions to Babel: HMAC-based authentication [RFC7298], source-specific routing [BABEL-SS], delay-based routing [BABEL-RTT] and ToS-specific routing [ToS-SPECIFIC]. 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 properly scoped and reasonably stable 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 6126bis.

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 6126bis. Use of other techniques for acquiring IPv4 connectivity (such as multiple layers of NAT) is strongly discouraged.

Rationale: support for IPv4 will likely 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-ietf-babel-source-specific [BABEL-SS].

Rationale: source-specific routing is an essential component of the Homenet architecture. 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 6126bis. 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. Appendix A of RFC 6126bis gives an example of a suitable algorithm.

Rationale: support for wireless transit links is a distinguishing feature of Homenet, and one that is requested by our users. 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".

While it would be desirable to perform link-quality probing on some wired link technologies, notably power-line networks, these kinds of links tend to be difficult or impossible to detect automatically, and we are not aware of any published link-quality algorithms for them. Hence, we do not require link-quality estimation for wired links of any kind.

2.2. Optional features

OPT1: a Homenet implementation of Babel MAY perform route selection by applying hysteresis to route metrics, as suggested in Section 3.6 of RFC 6126bis and described in detail in Section III.E of [BABEL-RTT]. However, hysteresis is not required, and the implementation 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.

OPT2: 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 configuration, which is done by HNCP, from 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 (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, in existing implementations HNCP installs 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 stack 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] 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. Optional features

OPT3: 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.

OPT4: 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 "Internal", "Ad Hoc" and "Hybrid" interfaces (Section 5.1 of [RFC7788]) that are assumed to be connected to links that are secured at a lower layer. HNCP and Babel packets are only accepted when they originate on these trusted links. "External" and "Guest" interfaces are connected to links that are not trusted, and any HNCP or Babel packets that are received on such interfaces are ignored. ("Leaf" interfaces are a special case, since they are connected to trusted links but HNCP and Babel traffic received on such interfaces is ignored.) This implies that the security of a Homenet network depends on the reliability of the border discovery procedure described in Section 5.3 of [RFC7788].

If untrusted links are used for transit, which is NOT RECOMMENDED, then any HNCP and Babel traffic that is carried over such links 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. IANA Considerations

This document requires no actions from IANA.

6. Acknowledgments

A number of people have helped with defining the requirements listed in this document. I am especially indebted to Barbara Stark and Markus Stenberg.

7. References

7.1. Normative References

[BABEL-SS] Boutier, M. and J. Chroboczek, "Source-Specific Routing in Babel", draft-ietf-babel-source-specific-03 (work in progress), August 2018.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997.

[RFC6126bis] Chroboczek, J. and D. Schinazi, "The Babel Routing Protocol", Internet Draft draft-ietf-babel-rfc6126bis-04, October 2017.

[RFC7788] Stenberg, M., Barth, S., and P. Pfister, "Home Networking Control Protocol", RFC 7788, DOI 10.17487/RFC7788, April 2016.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017.

7.2. 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.

[ToS-SPECIFIC] Chouasne, G., "https://tools.ietf.org/id/draft-chouasne-babel-tos-specific-00.xml", draft-chouasne-babel-tos-specific-00 (work in progress), July 2017.

Author's Address

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

Email: jch@irif.fr

HOMENET
Internet-Draft
Intended status: Standards Track
Expires: March 26, 2016

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

W. Cloetens
SoftAtHome

September 23, 2015

Outsourcing Home Network Authoritative Naming Service
draft-ietf-homenet-front-end-naming-delegation-04.txt

Abstract

RFC7368 'IPv6 Home Networking Architecture Principles' section 3.7 describes architecture principles related to naming and service discovery in residential home networks.

Customer Edge Routers and other Customer Premises Equipment (CPEs) are designed to provide IP connectivity to home networks. Most CPEs assign IP addresses to the nodes of the home network which makes them good candidates for hosting the naming service. IPv6 provides global connectivity, and nodes from the home network will be reachable from the global Internet. As a result, the naming service is expected to be exposed on the Internet.

However, CPEs have not been designed to host such a naming service exposed on the Internet. Running a naming service visible on the Internet may expose the CPEs 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.

In addition, regular end users may not understand, or possess the necessary skills to be able to perform, DNSSEC management and configuration. Misconfiguration may also result in naming service disruption, thus these end users may prefer to rely on third party name service providers.

This document describes a homenet naming architecture, where the CPEs manage the DNS zones associated with its own home network, and outsource elements of the naming service (possibly including DNSSEC management) to a third party running on the Internet.

Status of This Memo

Copyright Notice

Table of Contents

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. This makes CPEs good candidates to manage the binding between names and IP addresses of nodes. 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 the Homenet Zone, thus providing bindings between names and IP addresses. All this makes the CPE the natural candidate for populating the Homenet zone.

CPEs are usually low powered devices designed for the home network, but 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 CPEs, 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 Zone built by the CPE is outsourced to an Outsourcing Infrastructure. The Outsourcing Infrastructure publishes the corresponding Public Homenet Zone on the Internet. Section 4.1 describes the 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 CPE 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.

2.1. Scope of the Document

The scope of the document is to describe an architecture that outsources the authoritative naming service of the home network and more specifically the interactions between the home network and the Outsourcing Infrastructure. Considerations or descriptions on inner communications or organization of the home network are provided for completeness and for clarifying the interface between the home network and the Outsourcing Infrastructure

For sake of simplicity, this document designates by "CPE" that connects the home network with the Outsourcing Infrastructure - and so performs most of the operations for outsourcing the home network naming architecture. On the other hand, CPE are usually associated to home router. These two functions - e.g routing and naming - are not correlated and could be split in multiple devices. More

specifically, this document does not consider the home network has a single CPE nor a single ISP.

In fact this document considers the CPE as a set of functionalities that can be collocated on a single device or split between multiple devices. The split of these functions between different devices as well as their potential associated communications is considered as implementation dependent and out of scope of the architecture. The only limitation this architecture considers is that the Hidden Primary be located in a single place as long as the distributed Primary architecture has not been defined. As a result, if there are multiple candidates for hosting the Hidden Primary, the home network should select a single device.

3. Terminology

- Customer Premises Equipment: (CPE) is a router providing connectivity to the home network. It might be configured and managed by the end user. In this document, the CPE might also host services such as DHCPv6. This device might be provided by the ISP. A home network may have multiple CPE, and each of them may be connected to an Internet Service Provider. In this document, the CPE represents the set of functions involved in the naming architecture. These functions may be collocated on a single device, or distributed between multiple devices. How these functions communicate is out of the scope of this document and is left for implementation. See Section 2.1 for more details.

- 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 CPE and contains the bindings between names and IP addresses of the nodes in the home network. The CPE 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 CPE synchronizes the Homenet Zone. The Synchronization Server is configured as a secondary and the CPE 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 CPE 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 CPE synchronizes the Homenet Reverse Zone. It is configured as a secondary and the CPE 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 CPE signs the zone, the synchronization function might also be operated by the Reverse Public Authoritative Servers.

- Hidden Primary: designates the primary server of the CPE, that synchronizes the Homenet Zone with the Synchronization Server. A primary / secondary architecture is used between the CPE 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 CPE 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 CPE. It also lists the necessary parameters the CPE 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 CPE builds the Homenet Zone associated with the home network. How the Homenet Zone is built is out of the scope of this document. The CPE 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 CPE 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 CPE does not sign the Homenet Zone, the Outsourcing Infrastructure may instead sign it on behalf of the CPE. Figure 1 provides a more detailed description of the Outsourcing Infrastructure, but overall, it is expected that the CPE 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 CPE. Figure 1 illustrates the case of the resolution of node1.example.com.

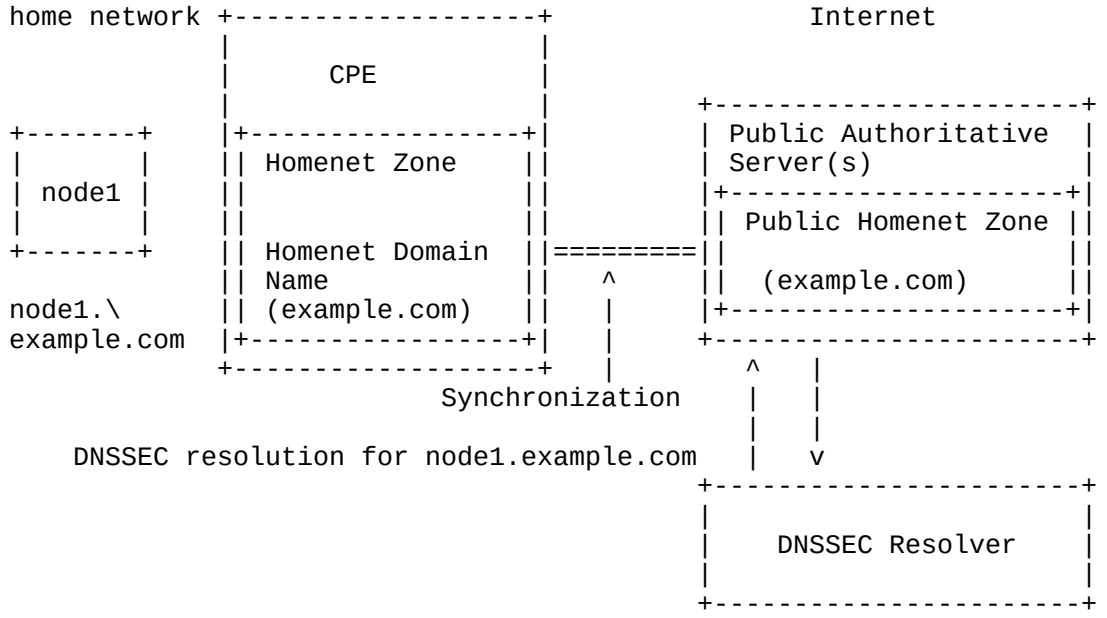


Figure 1: Homenet Naming Architecture Description

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 CPE outsources this operation to the Synchronization Server. On the other hand, if the CPE 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.

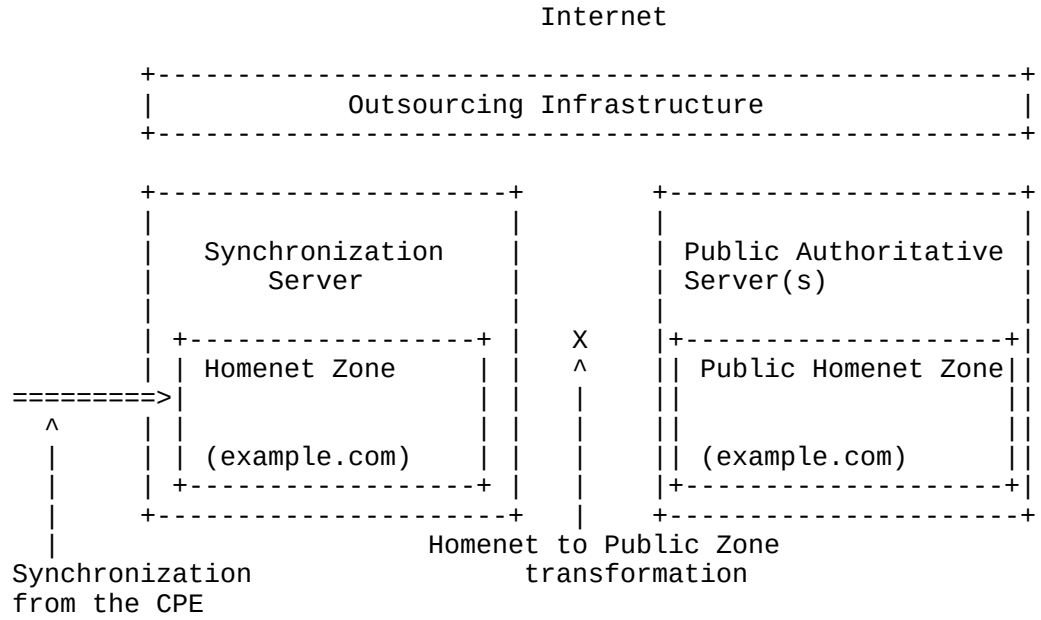


Figure 2: Outsourcing Infrastructure Description

4.2. Example: Homenet Zone

This section is not normative and intends to illustrate how the CPE 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 CPE. Motivations for keeping these two zones identical are detailed in Section 7, and this section considers that the CPE 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 CPE 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 CPE 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 CPE 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 CPE 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: CPE necessary parameters for outsourcing

This section specifies the various parameters required by the CPE to configure the naming architecture of this document. This section is informational, and is intended to clarify the information handled by the CPE 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 CPE 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 CPE 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 CPE 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 CPE for the primary / secondary configuration as well as to index the NOTIFY queries of the CPE when the CPE 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 CPE 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 CPE (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 CPEs are

candidate for hosting the Hidden Primary, some specific mechanisms should be designed so the home network only selects a single CPE for the Hidden Primary. Selection mechanisms based on HNCP are good candidates [XXX].

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 CPE 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 CPE 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 CPE SHOULD filter IXFR/ AXFR traffic and drop traffic not initiated by the Synchronization Server. The CPE 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 CPE 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 CPE 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 CPE, 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 CPE. 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 CPE. 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 CPE and the Synchronization Server. Similarly to TLS/DTLS, most CPEs 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 CPE 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 CPE is connected to the Internet, and IPsec communication between the CPE 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 CPE to manage a private key, a public key, or certificates, as well as Certificate Authorities. This adds complexity to the configuration especially on the CPE side. For this reason, we RECOMMEND that the CPE 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 CPE and the Synchronization Server SHOULD NOT use the external IP address of the CPE 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. CPE 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 CPE or by the Outsourcing Infrastructure. Section 6.1 details the implications of these two alternatives. Similarly, the secure delegation can be performed by the CPE 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 CPE or by the Outsourcing Infrastructure. It is RECOMMENDED that the CPE signs the zone unless there is a strong argument against this, such as a CPE that is not capable of signing the zone. In that case zone signing MAY be performed by the Outsourcing Infrastructure on behalf of the CPE.

Reasons for signing the zone by the CPE 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 CPE the ability to respond on behalf of the Public Authoritative Server(s). This could be useful for example if, in the future, the CPE announces to the home network that the CPE can act as a local authoritative primary or equivalent for the Homenet Zone. Currently the CPE 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 CPEs 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 CPE, it makes modification of the DNS data -- for example for flow redirection -- impossible. As a result, signing the Homenet Zone by the CPE provides better protection for end user privacy.

Reasons for signing the zone by the Outsourcing Infrastructure are:

- 1: The CPE 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 CPE and transparently for the end user. Avoiding DNSSEC management is mostly motivated by bad software implementations.

- 3: Reducing the impact of CPE replacement on the Public Homenet Zone. Unless the CPE private keys can be extracted and stored off-device, CPE 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 CPE and the Synchronization Server, authentication associations between the CPE and the Synchronization Server would need to be re-configured. As CPE 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 CPE. 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 CPE 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 CPE 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 CPE and the parent DNS server may be hard to maintain with CPEs, 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 CPE 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 CPE administration interface that provides the GUI to administer your CPE 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 CPE, 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 CPE 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 CPE 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 CPE 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 CPE, it may be able to authenticate the CPE 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 CPE 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 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_NEW the time the new IP address replaces the old one and T_OLD_UNREACHABLE the time the Synchronization Server is not reachable anymore with its old IP address. Seamless reachability is provided as long as T_OLD_UNREACHABLE - T_NEW > 2 * TTL. If this condition is not met, the Synchronization Server may be unreachable during 2 * TTL - (T_OLD_UNREACHABLE - T_NEW). In the case of a break-before-make, T_OLD_UNREACHABLE = T_NEW, and it may become unreachable up to 2 * 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 CPE 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 [I-D.ietf-opsec-ipv6-host-scanning].

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 CPE’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 CPE 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 CPE 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 CPE 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 CPE.

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 CPE, 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 CPE. This is done by outsourcing the DNS service to the Public Authoritative Server(s). By doing so, the CPE limits its DNS interactions between the Hidden Primary and the Synchronization Server. This limits the number of entities the CPE 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 CPE 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 CPE 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 CPE 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 CPE; 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>.

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., Cloetens, W., Griffiths, C., and R. Weber, "DHCP Options for Homenet Naming Architecture", draft-ietf-homenet-naming-architecture-dhc-options-02 (work in progress), May 2015.

[I-D.ietf-opsec-ipv6-host-scanning]
Gont, F. and T. Chown, "Network Reconnaissance in IPv6 Networks", draft-ietf-opsec-ipv6-host-scanning-08 (work in progress), August 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>.

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

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

Email: ralf.weber@nominum.com
URI: http://www.nominum.com

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

Email: v6ops@globis.net
URI: http://www.globis.net

Chris Griffiths

Email: cgriffiths@gmail.com

Wouter Cloetens
SoftAtHome
vaartdijk 3 701
3018 Wijgmaal
Belgium

Email: wouter.cloetens@softathome.com

Homenet Working Group
Internet-Draft
Obsoletes: 7788 (if approved)
Intended status: Standards Track
Expires: January 9, 2017

M. Stenberg
S. Barth
Independent
P. Pfister
Cisco Systems

July 8, 2016

Home Networking Control Protocol
draft-ietf-homenet-hncp-bis-00

Abstract

This document describes the Home Networking Control Protocol (HNCP), an extensible configuration protocol, and a set of requirements for home network devices. HNCP is described as a profile of and extension to the Distributed Node Consensus Protocol (DNCP). HNCP enables discovery of network borders, automated configuration of addresses, name resolution, service discovery, and the use of any routing protocol that supports routing based on both the source and destination address.

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

Table of Contents

1. Introduction

The Home Networking Control Protocol (HNCP) is designed to facilitate the sharing of state among home routers to fulfill the needs of the IPv6 homenet architecture [RFC7368], which assumes zero-configuration operation, multiple subnets, multiple home routers, and (potentially) multiple upstream service providers providing (potentially) multiple prefixes to the home network. While RFC 7368 sets no requirements for IPv4 support, HNCP aims to support the dual-stack mode of operation, and therefore the functionality is designed with that in mind. The state is shared as TLVs transported in the DNCP node state among the routers (and potentially advanced hosts) to enable:

- Autonomic discovery of network borders (Section 5.3) based on Distributed Node Consensus Protocol (DNCP) topology.
- Automated portioning of prefixes delegated by the service providers as well as assigned prefixes to both HNCP and non-HNCP routers (Section 6.3) using [RFC7695]. Prefixes assigned to HNCP routers are used to:
  - Provide addresses to non-HNCP aware nodes (using Stateless Address Autoconfiguration (SLAAC) and DHCP).
  - Provide space in which HNCP nodes assign their own addresses (Section 6.4).
- Internal and external name resolution, as well as multi-link service discovery (Section 8).
- Other services not defined in this document that do need to share state among homenet nodes and do not cause rapid and constant TLV changes (see the following applicability section).

HNCP is a protocol based on DNCP [RFC7787] and includes a DNCP profile that defines transport and synchronization details for sharing state across nodes defined in Section 3. The rest of the

document defines behavior of the services noted above, how the required TLVs are encoded (Section 10), as well as additional requirements on how HNCP nodes should behave (Section 11).

1.1. Applicability

While HNCP does not deal with routing protocols directly (except potentially informing them about internal and external interfaces if classification specified in Section 5.3 is used), in homenet environments where multiple IPv6 source prefixes can be present, routing based on the source and destination address is necessary [RFC7368]. Ideally, the routing protocol is also zero configuration (e.g., no need to configure identifiers or metrics), although HNCP can also be used with a manually configured routing protocol.

As HNCP uses DNCP as the actual state synchronization protocol, the applicability statement of DNCP applies here as well; HNCP should not be used for any data that changes rapidly and constantly. If such data needs to be published in an HNCP network, 1) a more applicable protocol should be used for those portions, and 2) locators to a server of said protocol should be announced using HNCP instead. An example for this is naming and service discovery (Section 8) for which HNCP only transports DNS server addresses and no actual per-name or per-service data of hosts.

HNCP TLVs specified within this document, in steady state, stay constant, with one exception: as Delegated-Prefix TLVs (Section 10.2.1) do contain lifetimes, they force republishing of that data every time the valid or preferred lifetimes of prefixes are updated (significantly). Therefore, it is desirable for ISPs to provide large enough valid and preferred lifetimes to avoid unnecessary HNCP state churn in homes, but even given non-cooperating ISPs, the state churn is proportional only to the number of externally received delegated prefixes and not to the home network size, and it should therefore be relatively low.

HNCP assumes a certain level of control over host configuration servers (e.g., DHCP [RFC2131]) on links that are managed by its routers. Some HNCP functionality (such as border discovery or some aspects of naming) might be affected by existing DHCP servers that are not aware of the HNCP-managed network and thus might need to be reconfigured to not result in unexpected behavior.

While HNCP routers can provide configuration to and receive configuration from non-HNCP routers, they are not able to traverse such devices based solely on the protocol as defined in this document, i.e., HNCP routers that are connected only by different

interfaces of a non-HNCP router will not be part of the same HNCP network.

While HNCP is designed to be used by (home) routers, it can also be used by advanced hosts that want to do, e.g., their own address assignment and routing.

HNCP is link-layer agnostic; if a link supports IPv6 (link-local) multicast and unicast, HNCP will work on it. Trickle retransmissions and keep-alives will handle both packet loss and non-transitive connectivity, ensuring eventual convergence.

2. Terminology

The following terms are used as they are defined in [RFC7695]:

- Advertised Prefix Priority
- Advertised Prefix
- Assigned Prefix
- Delegated Prefix
- Prefix Adoption
- Private Link
- Published Assigned Prefix
- Applied Assigned Prefix
- Shared Link

The following terms are used as they are defined in [RFC7787]:

- DNCP profile
- Node identifier
- Link
- Interface

(HNCP) node	a device implementing this specification.
(HNCP) router	a device implementing this specification, which forwards traffic on behalf of other devices.
Greatest node identifier	when comparing the DNCP node identifiers of multiple nodes, the one that has the greatest value in a bitwise comparison.
Border	separation point between administrative domains; in this case, between the home network and any other network, i.e., usually an ISP network.
Internal link	a link that does not cross borders.
Internal interface	an interface that is connected to an internal link.
External interface	an interface that is connected to a link that is not an internal link.
Interface category	a local configuration denoting the use of a particular interface. The Interface category determines how an HNCP node should treat the particular interface. The External and Internal categories mark the interface as out of or within the network border; there are also a number of subcategories of Internal that further affect local node behavior. See Section 5.1 for a list of interface categories and how they behave. The Internal or External category may also be auto-detected (Section 5.3).
Border router	a router announcing external connectivity and forwarding traffic across the network border.
Common Link	a set of nodes on a link that share a common view of it, i.e., they see each other’s traffic and the same set of hosts. Unless configured otherwise, transitive connectivity is assumed.
DHCPv4	refers to the Dynamic Host Configuration Protocol [RFC2131] in this document.
DHCPv6	refers to the Dynamic Host Configuration Protocol for IPv6 (DHCPv6) [RFC3315] in this document.
DHCP	refers to cases that apply to both DHCPv4 and DHCPv6 in this document.

2.1. 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 RFC 2119 [RFC2119].

3. DNCP Profile

The DNCP profile for HNCP is defined as follows:

- HNCP uses UDP datagrams on port 8231 as a transport over link-local scoped IPv6, using unicast and multicast (FF02:0:0:0:0:0:0:11 is the HNCP group address). Received datagrams where either or both of the IPv6 source or destination addresses are not link-local scoped MUST be ignored. Replies to multicast and unicast messages MUST be sent to the IPv6 source address and port of the original message. Each node MUST be able to receive (and potentially reassemble) UDP datagrams with a payload of at least 4000 bytes.

- HNCP operates on multicast-capable interfaces only. HNCP nodes MUST assign a non-zero 32-bit endpoint identifier to each interface for which HNCP is enabled. The value 0 is not used in DNCP TLVs but has a special meaning in HNCP TLVs (see Sections 6.4 and 10.3). These identifiers MUST be locally unique within the scope of the node, and using values equivalent to the IPv6 link-local scope identifiers for the given interfaces are RECOMMENDED.

- HNCP uses opaque 32-bit node identifiers (DNCP_NODE_IDENTIFIER_LENGTH = 32). A node implementing HNCP SHOULD use a random node identifier. If there is a node identifier collision (as specified in the Node-State TLV handling of Section 4.4 of [RFC7787]), the node MUST immediately generate and use a new random node identifier that is not used by any other node at the time, based on the current DNCP network state.

- HNCP nodes MUST use the leading 64 bits of the MD5 message digest [RFC1321] as the DNCP hash function H(x) used in building the DNCP hash tree.

- HNCP nodes MUST use DNCP’s per-endpoint keep-alive extension on all endpoints. The following parameters are suggested:

  * Default keep-alive interval (DNCP_KEEPALIVE_INTERVAL): 20 seconds.

    * Multiplier (DNCP_KEEPALIVE_MULTIPLIER): 2.1 on virtually lossless links works fine, as it allows for one lost keep-alive. If used on a lossy link, a considerably higher multiplier, such as 15, should be used instead. In that case, an implementation might prefer shorter keep-alive intervals on that link as well to ensure that the timeout (equal to DNCP_KEEPALIVE_INTERVAL * DNCP_KEEPALIVE_MULTIPLIER) after which entirely lost nodes time out is low enough.

- HNCP nodes use the following Trickle parameters for the per-interface Trickle instances:

    * k SHOULD be 1, as the timer reset when data is updated, and further retransmissions should handle packet loss. Even on a non-transitive lossy link, the eventual per-endpoint keep-alives should ensure status synchronization occurs.

    * Imin SHOULD be 200 milliseconds but MUST NOT be lower. Note: earliest transmissions may occur at Imin / 2.

    * Imax SHOULD be 7 doublings of Imin [RFC6206] but MUST NOT be lower.

- HNCP unicast traffic SHOULD be secured using Datagram Transport Layer Security (DTLS) [RFC6347] as described in DNCP if exchanged over unsecured links. UDP on port 8232 is used for this purpose. A node implementing HNCP security MUST support the DNCP Pre-Shared Key (PSK) method, SHOULD support the PKI-based trust method, and MAY support the DNCP certificate-based trust consensus method. [RFC7525] provides guidance on how to securely utilize DTLS.

- HNCP nodes MUST ignore all Node-State TLVs received via multicast on a link that has DNCP security enabled in order to prevent spoofing of node state changes.

4. HNCP Versioning and Router Capabilities

Multiple versions of HNCP based on compatible DNCP profiles may be present in the same network when transitioning between HNCP versions, and for troubleshooting purposes, it might be beneficial to identify the HNCP agent version running. Therefore, each node MUST include an HNCP-Version TLV (Section 10.1) indicating the currently supported version in its node data and MUST ignore (except for DNCP synchronization purposes) any TLVs that have a type greater than 32 and that are published by nodes that didn’t also publish an HNCP-Version TLV.

HNCP routers may also have different capabilities regarding interactions with hosts, e.g., for configuration or service discovery. These are indicated by M, P, H, and L values. The combined "capability value" is a metric indicated by interpreting the bits as an integer, i.e., (M << 12 | P << 8 | H << 4 | L). These values are used to elect certain servers on a Common Link, as described in Section 7. Nodes that are not routers MUST announce the value 0 for all capabilities. Any node announcing the value 0 for a capability is considered to not advertise said capability and thus does not take part in the respective election.

5. Interface Classification

5.1. Interface Categories

HNCP specifies the following categories that interfaces can be configured to be in:

Internal category: This declares an interface to be internal, i.e., within the borders of the HNCP network. The interface MUST operate as a DNCP endpoint. Routers MUST forward traffic with appropriate source addresses between their internal interfaces and allow internal traffic to reach external networks. All nodes MUST implement this category, and nodes not implementing any other category implicitly use it as a fixed default.

External category: This declares an interface to be external, i.e., not within the borders of the HNCP network. The interface MUST NOT operate as a DNCP endpoint. Accessing internal resources from external interfaces is restricted, i.e., the use of Recommended Simple Security Capabilities in Customer Premises Equipments (CPEs) [RFC6092] is RECOMMENDED. HNCP routers SHOULD announce acquired configuration information for use in the network as described in Section 6.2, if the interface appears to be connected to an external network. HNCP routers MUST implement this category.

Leaf category: This declares an interface used by client devices only. Such an interface uses the Internal category with the exception that it MUST NOT operate as a DNCP endpoint. This category SHOULD be supported by HNCP routers.

Guest category: This declares an interface used by untrusted client devices only. In addition to the restrictions of the Leaf category, HNCP routers MUST filter traffic from and to the interface such that connected devices are unable to reach other devices inside the HNCP network or query services advertised by

them unless explicitly allowed. This category SHOULD be supported by HNCP routers.

Ad Hoc category: This configures an interface to use the Internal category, but no assumption is made about the link's transitivity. All other interface categories assume transitive connectivity. This affects the Common Link (Section 6.1) definition. Support for this category is OPTIONAL.

Hybrid category: This declares an interface to use the Internal category while still trying to acquire (external) configuration information on it, e.g., by running DHCP clients. This is useful, e.g., if the link is shared with a non-HNCP router under control and still within the borders of the same network. Detection of this category automatically in addition to manual configuration is out of scope of this document. Support for this category is OPTIONAL.

5.2. DHCP-Aided Auto-Detection

Auto-detection of interface categories is possible based on interaction with DHCPv4 [RFC2131] and DHCPv6 Prefix Delegation (DHCPv6-PD) [RFC3633] servers on connected links. HNCP defines special DHCP behavior to differentiate its internal servers from external ones in order to achieve this. Therefore, all internal devices (including HNCP nodes) running DHCP servers on links where auto-detection is used by any HNCP node MUST use the following mechanism based on "The User Class Option for DHCP" [RFC3004] and its DHCPv6 counterpart [RFC3315]:

- The device MUST ignore or reject DHCP-Requests containing a DHCP user class consisting of the ASCII string "HOMENET".

Not following this rule (e.g., running unmodified DHCP servers) might lead to false positives when auto-detection is used, i.e., HNCP nodes assume an interface to not be internal, even though it was intended to be.

5.3. Algorithm for Border Discovery

This section defines the interface classification algorithm. It is suitable for both IPv4 and IPv6 (single or dual stack) and detects the category of an interface either automatically or based on a fixed configuration. By determining the category for all interfaces, the network borders are implicitly defined, i.e., all interfaces not belonging to the External category are considered to be within the borders of the network; all others are not.

The following algorithm MUST be implemented by any node implementing HNCP. However, if the node does not implement auto-detection, only the first and last step are required. The algorithm works as follows, with evaluation stopping at first match:

1. If a fixed category is configured for an interface, it is used.

2. If a delegated prefix could be acquired by running a DHCPv6 client, it is considered external. The DHCPv6 client MUST have included a DHCPv6 user class consisting of the ASCII string "HOMENET" in all of its requests.

3. If an IPv4 address could be acquired by running a DHCPv4 client on the interface, it is considered external. The DHCPv4 client MUST have included a DHCP user class consisting of the ASCII string "HOMENET" in all of its requests.

4. The interface is considered internal.

Note that as other HNCP nodes will ignore the client due to the User Class option, any server that replies is clearly external (or a malicious internal node).

An HNCP router SHOULD allow setting the fixed category for each interface that may be connected to either an internal or external device (e.g., an Ethernet port that can be connected to a modem, another HNCP router, or a client). Note that all fixed categories except internal and external cannot be auto-detected and can only be selected using manual configuration.

An HNCP router using auto-detection on an interface MUST run the appropriately configured DHCP clients as long as the interface without a fixed category is active (including states where auto-detection considers it to be internal) and rerun the algorithm above to react to conditions resulting in a different interface category. The router SHOULD wait for a reasonable time period (5 seconds as a default), during which the DHCP clients can acquire a lease, before treating a newly activated or previously external interface as internal.

6. Autonomous Address Configuration

This section specifies how HNCP nodes configure host and node addresses. At first, border routers share information obtained from service providers or local configuration by publishing one or more External-Connection TLVs (Section 10.2). These contain other TLVs such as Delegated-Prefix TLVs (Section 10.2.1) that are then used for prefix assignment. Finally, HNCP nodes obtain addresses either

statelessly or using a specific stateful mechanism (Section 6.4). Hosts and non-HNCP routers are configured using SLAAC, DHCP, or DHCPv6-PD.

6.1. Common Link

HNCP uses the concept of Common Link both in autonomic address configuration and naming and service discovery (Section 8). A Common Link refers to the set of interfaces of nodes that see each other's traffic and presumably also the traffic of all hosts that may use the nodes to, e.g., forward traffic. Common Links are used, e.g., to determine where prefixes should be assigned or which peers participate in the election of a DHCP server. The Common Link is computed separately for each local internal interface, and it always contains the local interface. Additionally, if the local interface is not set to the Ad Hoc category (see Section 5.1), it also contains the set of interfaces that are bidirectionally reachable from the given local interface; that is, every remote interface of a remote node meeting all of the following requirements:

- The local node publishes a Peer TLV with:
  - Peer Node Identifier = remote node's node identifier
  - Peer Endpoint Identifier = remote interface's endpoint identifier
  - Endpoint Identifier = local interface's endpoint identifier
- The remote node publishes a Peer TLV with:
  - Peer Node Identifier = local node's node identifier
  - Peer Endpoint Identifier = local interface's endpoint identifier
  - Endpoint Identifier = remote interface's endpoint identifier

A node MUST be able to detect whether two of its local internal interfaces are connected, e.g., by detecting an identical remote interface being part of the Common Links of both local interfaces.

6.2. External Connections

Each HNCP router MAY obtain external connection information such as address prefixes, DNS server addresses, and DNS search paths from one or more sources, e.g., DHCPv6-PD [RFC3633], NETCONF [RFC6241], or static configuration. Each individual external connection to be

shared in the network is represented by one External-Connection TLV (Section 10.2).

Announcements of individual external connections can consist of the following components:

Delegated Prefixes: Address space available for assignment to internal links announced using Delegated-Prefix TLVs (Section 10.2.1). Some address spaces might have special properties that are necessary to understand in order to handle them (e.g., information similar to [RFC6603]). This information is encoded using DHCPv6-Data TLVs (Section 10.2.2) inside the respective Delegated-Prefix TLVs.

Auxiliary Information: Information about services such as DNS or time synchronization regularly used by hosts in addition to addressing and routing information. This information is encoded using DHCPv6-Data TLVs (Section 10.2.2) and DHCPv4-Data TLVs (Section 10.2.3).

Whenever information about reserved parts (e.g., as specified in [RFC6603]) is received for a delegated prefix, the reserved parts MUST be advertised using Assigned-Prefix TLVs (Section 10.3) with the greatest priority (i.e., 15), as if they were assigned to a Private Link.

Some connections or delegated prefixes may have a special meaning and are not regularly used for internal or Internet connectivity; instead, they may provide access to special services like VPNs, sensor networks, Voice over IP (VoIP), IPTV, etc. Care must be taken that these prefixes are properly integrated and dealt with in the network, in order to avoid breaking connectivity for devices who are not aware of their special characteristics or to only selectively allow certain devices to use them. Such prefixes are distinguished using Prefix-Policy TLVs (Section 10.2.1.1). Their contents MAY be partly opaque to HNCP nodes, and their identification and usage depends on local policy. However, the following general rules MUST be adhered to:

Special rules apply when making address assignments for prefixes with Prefix-Policy TLVs with type 131, as described in Section 6.3.2.

In the presence of any type 1 to 128 Prefix-Policy TLV, the prefix is specialized to reach destinations denoted by any such Prefix-Policy TLV, i.e., in absence of a type 0 Prefix-Policy TLV, it is not usable for general Internet connectivity. An HNCP router MAY enforce this restriction with appropriate packet filter rules.

6.3. Prefix Assignment

HNCP uses the prefix assignment algorithm [RFC7695] in order to assign prefixes to HNCP internal links and uses some of the terminology (Section 2) defined there. HNCP furthermore defines the Assigned-Prefix TLV (Section 10.3), which MUST be used to announce Published Assigned Prefixes.

6.3.1. Prefix Assignment Algorithm Parameters

All HNCP nodes running the prefix assignment algorithm use the following values for its parameters:

Node IDs: HNCP node identifiers are used. The comparison operation is defined as bitwise comparison.

Set of Delegated Prefixes: The set of prefixes encoded in Delegated-Prefix TLVs that are not strictly included in prefixes encoded in other Delegated-Prefix TLVs. Note that Delegated-Prefix TLVs included in ignored External-Connection TLVs are not considered. It is dynamically updated as Delegated-Prefix TLVs are added or removed.

Set of Shared Links: The set of Common Links associated with interfaces with the Internal, Leaf, Guest, or Ad Hoc category. It is dynamically updated as interfaces are added, removed, or switched from one category to another. When multiple interfaces are detected as belonging to the same Common Link, prefix assignment is disabled on all of these interfaces except one.

Set of Private Links: This document defines Private Links as representing DHCPv6-PD clients or as a mean to advertise prefixes included in the DHCPv6 Exclude Prefix option. Other implementation-specific Private Links may be defined whenever a prefix needs to be assigned for a purpose that does not require a consensus with other HNCP nodes.

Set of Advertised Prefixes: The set of prefixes included in Assigned-Prefix TLVs advertised by other HNCP nodes (prefixes advertised by the local node are not in this set). The associated Advertised Prefix Priority is the priority specified in the TLV. The associated Shared Link is determined as follows:

* If the Link Identifier is 0, the Advertised Prefix is not assigned on a Shared Link.

* If the other node’s interface identified by the Link Identifier is included in one of the Common Links used for prefix

assignment, it is considered as assigned on the given Common Link.

* Otherwise, the Advertised Prefix is not assigned on a Shared Link.

Advertised Prefixes as well as their associated priorities and associated Shared Links MUST be updated as Assigned-Prefix TLVs are added, updated, or removed, and as Common Links are modified.

ADOPT_MAX_DELAY: The default value is 0 seconds (i.e., prefix adoption is done instantly).

BACKOFF_MAX_DELAY: The default value is 4 seconds.

RANDOM_SET_SIZE: The default value is 64.

Flooding Delay: The default value is 5 seconds.

Default Advertised Prefix Priority: When a new assignment is created or an assignment is adopted -- as specified in the prefix assignment algorithm routine -- the default Advertised Prefix Priority to be used is 2.

6.3.2. Making New Assignments

Whenever the prefix assignment algorithm subroutine (Section 4.1 of [RFC7695]) is run on a Common Link, and whenever a new prefix may be assigned (case 1 of the subroutine: no Best Assignment and no Current Assignment), the decision of whether the assignment of a new prefix is desired MUST follow these rules in order:

If the Delegated-Prefix TLV contains a DHCPv6-Data TLV, and the meaning of one of the DHCP options is not understood by the HNCP node, the creation of a new prefix is not desired. This rule applies to TLVs inside Delegated-Prefix TLVs but not to those inside External-Connection TLVs.

If the remaining preferred lifetime of the prefix is 0 and there is another delegated prefix of the same IP version used for prefix assignment with a non-zero preferred lifetime, the creation of a new prefix is not desired.

If the Delegated-Prefix TLV does not include a Prefix-Policy TLV indicating restrictive assignment (type 131) or if local policy exists to identify it based on, e.g., other Prefix-Policy TLV values and allows assignment, the creation of a new prefix is desired.

Otherwise, the creation of a new prefix is not desired.

If the considered delegated prefix is an IPv6 prefix, and whenever there is at least one available prefix of length 64, a prefix of length 64 MUST be selected unless configured otherwise. In case no prefix of length 64 would be available, a longer prefix MAY be selected even without configuration.

If the considered delegated prefix is an IPv4 prefix (Section 6.5 details how IPv4-delegated prefixes are generated), a prefix of length 24 SHOULD be preferred.

In any case, an HNCP router making an assignment MUST support a mechanism suitable to distribute addresses from the considered prefix if the link is intended to be used by clients. In this case, a router assigning an IPv4 prefix MUST announce the L-capability, and a router assigning an IPv6 prefix with a length greater than 64 MUST announce the H-capability as defined in Section 4.

6.3.3. Applying Assignments

The prefix assignment algorithm indicates when a prefix is applied to the respective Common Link. When that happens, each router connected to said link:

MUST forward traffic destined to said prefix to the respective link.

MUST participate in the client configuration election as described in Section 7, if the link is intended to be used by clients.

MAY add an address from said prefix to the respective network interface as described in Section 6.4, e.g., if it is to be used as source for locally originating traffic.

6.3.4. DHCPv6 Prefix Delegation

When an HNCP router announcing the P-Capability (Section 4) receives a DHCPv6-PD request from a client, it SHOULD assign one prefix per delegated prefix in the network. This set of assigned prefixes is then delegated to the client, after it has been applied as described in the prefix assignment algorithm. Each DHCPv6-PD client MUST be considered as an independent Private Link, and delegation MUST be based on the same set of delegated prefixes as the one used for Common Link prefix assignments; however, the prefix length to be delegated MAY be smaller than 64.

The assigned prefixes MUST NOT be given to DHCPv6-PD clients before they are applied and MUST be withdrawn whenever they are destroyed. As an exception to this rule, in order to shorten delays of processed requests, a router MAY prematurely give out a prefix that is advertised but not yet applied if it does so with a valid lifetime of not more than 30 seconds and ensures removal or correction of lifetimes as soon as possible.

6.4. Node Address Assignment

This section specifies how HNCP nodes reserve addresses for their own use. Nodes MAY, at any time, try to reserve a new address from any Applied Assigned Prefix. Each HNCP node SHOULD announce an IPv6 address and -- if it supports IPv4 -- MUST announce an IPv4 address, whenever matching prefixes are assigned to at least one of its Common Links. These addresses are published using Node-Address TLVs and used to locally reach HNCP nodes for other services. Nodes SHOULD NOT create and announce more than one assignment per IP version to avoid cluttering the node data with redundant information unless a special use case requires it.

Stateless assignment based on Semantically Opaque Interface Identifiers [RFC7217] SHOULD be used for address assignment whenever possible (e.g., the prefix length is 64), otherwise (e.g., for IPv4 if supported) the following method MUST be used instead: For any assigned prefix for which stateless assignment is not used, the first quarter of the addresses are reserved for HNCP-based address assignments, whereas the last three quarters are left to the DHCP elected router (Section 4 specifies the DHCP server election process). For example, if the prefix 192.0.2.0/24 is assigned and applied to a Common Link, addresses included in 192.0.2.0/26 are reserved for HNCP nodes, and the remaining addresses are reserved for the elected DHCPv4 server.

HNCP nodes assign addresses to themselves and then (to ensure eventual lack of conflicting assignments) publish the assignments using the Node-Address TLV (Section 10.4).

The process of obtaining addresses is specified as follows:

- A node MUST NOT start advertising an address if it is already advertised by another node.

- An assigned address MUST be part of an assigned prefix currently applied on a Common Link that includes the interface specified by the endpoint identifier.

- An address MUST NOT be used unless it has been advertised for at least ADDRESS_APPLY_DELAY consecutive seconds and is still currently being advertised. The default value for ADDRESS_APPLY_DELAY is 3 seconds.

- Whenever the same address is advertised by more than one node, all but the one advertised by the node with the greatest node identifier MUST be removed.

6.5. Local IPv4 and ULA Prefixes

HNCP routers can create a Unique Local Address (ULA) or private IPv4 prefix to enable connectivity between local devices. These prefixes are inserted in HNCP as if they were delegated prefixes of a (virtual) external connection (Section 6.2). The following rules apply:

> An HNCP router SHOULD create a ULA prefix if there is no other IPv6 prefix with a preferred time greater than 0 in the network. It MAY also do so if there are other delegated IPv6 prefixes, but none of which is locally generated (i.e., without any Prefix-Policy TLV) and has a preferred time greater than 0. However, it MUST NOT do so otherwise. In case multiple locally generated ULA prefixes are present, only the one published by the node with the greatest node identifier is kept among those with a preferred time greater than 0 -- if there is any.
>
> An HNCP router MUST create a private IPv4 prefix [RFC1918] whenever it wishes to provide IPv4 Internet connectivity to the network and no other private IPv4 prefix with Internet connectivity currently exists. It MAY also enable local IPv4 connectivity by creating a private IPv4 prefix if no IPv4 prefix exists but MUST NOT do so otherwise. In case multiple IPv4 prefixes are announced, only the one published by the node with the greatest node identifier is kept among those with a Prefix-Policy TLV of type 0 -- if there is any. The router publishing a prefix with Internet connectivity MUST forward IPv4 traffic to the Internet and perform NAT on behalf of the network as long as it publishes the prefix; other routers in the network MAY choose not to.

Creation of such ULA and IPv4 prefixes MUST be delayed by a random time span between 0 and 10 seconds in which the router MUST scan for others trying to do the same.

When a new ULA prefix is created, the prefix is selected based on the configuration, using the last non-deprecated ULA prefix, or generated based on [RFC4193].

7. Configuration of Hosts and Non-HNCP Routers

HNCP routers need to ensure that hosts and non-HNCP downstream routers on internal links are configured with addresses and routes. Since DHCP clients can usually only bind to one server at a time, a per-link and per-service election takes place.

HNCP routers may have different capabilities for configuring downstream devices and providing naming services. Each router MUST therefore indicate its capabilities as specified in Section 4 in order to participate as a candidate in the election.

7.1. IPv6 Addressing and Configuration

In general, Stateless Address Autoconfiguration [RFC4861] is used for client configuration for its low overhead and fast renumbering capabilities. Therefore, each HNCP router sends Router Advertisements on interfaces that are intended to be used by clients and MUST at least include a Prefix Information Option for each Applied Assigned Prefix that it assigned to the respective link in every such advertisement. However, stateful DHCPv6 can be used in addition by administrative choice to, e.g., collect hostnames and use them to provide naming services or whenever stateless configuration is not applicable.

The designated stateful DHCPv6 server for a Common Link (Section 6.1) is elected based on the capabilities described in Section 4. The winner is the router (connected to the Common Link) advertising the greatest H-capability. In case of a tie, Capability Values (Section 4) are compared, and the router with the greatest value is elected. In case of another tie, the router with the greatest node identifier is elected among the routers with tied Capability Values.

The elected router MUST serve stateful DHCPv6 and SHOULD provide naming services for acquired hostnames as outlined in Section 8; all other nodes MUST NOT. Stateful addresses SHOULD be assigned in a way that does not hinder fast renumbering even if the DHCPv6 server or client do not support the DHCPv6 reconfigure mechanism, e.g., by only handing out leases from locally generated (ULA) prefixes and prefixes with a length different from 64 and by using low renew and rebind times (i.e., not longer than 5 minutes). In case no router was elected, stateful DHCPv6 is not provided. Routers that cease to be elected DHCP servers SHOULD -- when applicable -- invalidate remaining existing bindings in order to trigger client reconfiguration.

7.2. DHCPv6 for Prefix Delegation

The designated DHCPv6 server for prefix delegation on a Common Link is elected based on the capabilities described in Section 4. The winner is the router (connected to the Common Link) advertising the greatest P-capability. In case of a tie, Capability Values (Section 4) are compared, and the router with the greatest value is elected. In case of another tie, the router with the greatest node identifier is elected among the routers with tied Capability Values.

The elected router MUST provide prefix delegation services [RFC3633] on the given link (and follow the rules in Section 6.3.4); all other nodes MUST NOT.

7.3. DHCPv4 for Addressing and Configuration

The designated DHCPv4 server on a Common Link (Section 6.1) is elected based on the capabilities described in Section 4. The winner is the router (connected to the Common Link) advertising the greatest L-capability. In case of a tie, Capability Values (Section 4) are compared, and the router with the greatest value is elected. In case of another tie, the router with the greatest node identifier is elected among the routers with tied Capability Values.

The elected router MUST provide DHCPv4 services on the given link; all other nodes MUST NOT. The elected router MUST provide IP addresses from the pool defined in Section 6.4 and MUST announce itself as router [RFC2132] to clients.

DHCPv4 lifetimes renew and rebind times (T1 and T2) SHOULD be short (i.e., not longer than 5 minutes) in order to provide reasonable response times to changes. Routers that cease to be elected DHCP servers SHOULD -- when applicable -- invalidate remaining existing bindings in order to trigger client reconfiguration.

7.4. Multicast DNS Proxy

The designated Multicast DNS (mDNS) [RFC6762] proxy on a Common Link is elected based on the capabilities described in Section 4. The winner is the router (connected to the Common Link) advertising the greatest M-capability. In case of a tie, Capability Values (Section 4) are compared, and the router with the greatest value is elected. In case of another tie, the router with the greatest node identifier is elected among the routers with tied Capability Values.

The elected router MUST provide an mDNS proxy on the given link and announce it as described in Section 8.

8. Naming and Service Discovery

Network-wide naming and service discovery can greatly improve the user friendliness of a network. The following mechanism provides means to setup and delegate naming and service discovery across multiple HNCP routers.

Each HNCP router SHOULD provide and advertise a recursive name resolving server to clients that honor the announcements made in Delegated-Zone TLVs (Section 10.5), Domain-Name TLVs (Section 10.6), and Node-Name TLVs (Section 10.7), i.e., delegate queries to the designated name servers and hand out appropriate A, AAAA, and PTR records according to the mentioned TLVs.

Each HNCP router SHOULD provide and announce an auto-generated or user-configured name for each internal Common Link (Section 6.1) for which it is the designated DHCPv4, stateful DHCPv6 server, mDNS proxy, or for which it provides forward or reverse DNS services on behalf of connected devices. This announcement is done using Delegated-Zone TLVs (Section 10.5) and MUST be unique in the whole network. In case of a conflict, the announcement of the node with the greatest node identifier takes precedence, and all other nodes MUST cease to announce the conflicting TLV. HNCP routers providing recursive name resolving services MUST use the included DNS server address within the TLV to resolve names belonging to the zone as if there was an NS record.

Each HNCP node SHOULD announce a node name for itself to be easily reachable and MAY announce names on behalf of other devices. Announcements are made using Node-Name TLVs (Section 10.7), and the announced names MUST be unique in the whole network. In case of a conflict, the announcement of the node with the greatest node identifier takes precedence, and all other nodes MUST cease to announce the conflicting TLV. HNCP routers providing recursive name resolving services as described above MUST resolve such announced names to their respective IP addresses as if there were corresponding A/AAAA records.

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. A default value for this TLV MUST be set, although the default value of the Domain-Name TLV (Section 10.6) is out of scope of this document, and an administrator MAY configure the announcement of a Domain-Name TLV for the network. In case multiple are announced, the domain of the node with the greatest node identifier takes precedence.

9. Securing Third-Party Protocols

PSKs are often required to secure (for example) IGPs and other protocols that lack support for asymmetric security. The following mechanism manages PSKs using HNCP to enable bootstrapping of such third-party protocols. The scheme SHOULD NOT be used unless it's in conjunction with secured HNCP unicast transport (i.e., DTLS), as transferring the PSK in plaintext anywhere in the network is a potential risk, especially as the originator may not know about security (and use of DNCP security) on all links. The following rules define how such a PSK is managed and used:

- If no Managed-PSK TLV (Section 10.8) is currently being announced, an HNCP node using this mechanism MUST create one after a random delay of 0 to 10 seconds with a 32 bytes long random key and add it to its node data.

- In case multiple nodes announce such a TLV at the same time, all but the one with the greatest node identifier stop advertising it and adopt the remaining one.

- The node currently advertising the Managed-PSK TLV MUST generate and advertise a new random one whenever an unreachable node is removed from the DNCP topology as described in Section 4.6 of [RFC7787].

PSKs for individual protocols SHOULD be derived from the random PSK using a suitable one-way hashing algorithm (e.g., by using the HMAC-based Key Derivation Function (HKDF) based on HMAC-SHA256 [RFC6234] with the particular protocol name in the info field) so that disclosure of any derived key does not impact other users of the managed PSK. Furthermore, derived PSKs MUST be updated whenever the managed PSK changes.

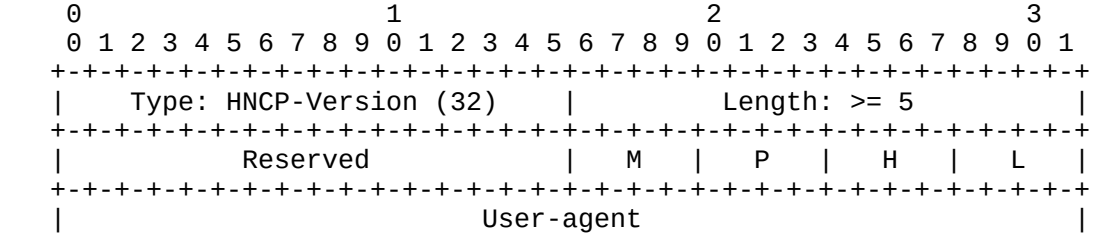
10. Type-Length-Value Objects

HNCP defines the following TLVs in addition to those defined by DNCP. The same general rules and defaults for encoding as noted in Section 7 of [RFC7787] apply. Note that most HNCP variable-length TLVs also support optional nested TLVs, and they are encoded after the variable-length content, followed by the zero padding of the variable-length content to the next 32-bit boundary.

TLVs defined here are only valid when appearing in their designated context, i.e., only directly within container TLVs mentioned in their definition or -- absent any mentions -- only as top-level TLVs within the node data set. TLVs appearing outside their designated context MUST be ignored.

TLVs encoding IP addresses or prefixes allow encoding both IPv6 and IPv4 addresses and prefixes. IPv6 information is encoded as is, whereas for IPv4, the IPv4-mapped IPv6 addresses format [RFC4291] is used, and prefix lengths are encoded as the original IPv4 prefix length increased by 96.

10.1. HNCP-Version TLV



This TLV is used to indicate the supported version and router capabilities of an HNCP node as described in Section 4.

Reserved: Bits are reserved for future use. They MUST be set to 0 when creating this TLV, and their value MUST be ignored when processing the TLV.

M-capability: Priority value used for electing the on-link mDNS [RFC6762] proxy. It MUST be set to 0 if the router is not capable of proxying mDNS, otherwise it SHOULD be set to 4 but MAY be set to any value from 1 to 7 to indicate a non-default priority. The values 8-15 are reserved for future use.

P-capability: Priority value used for electing the on-link DHCPv6-PD server. It MUST be set to 0 if the router is not capable of providing prefixes through DHCPv6-PD (Section 6.3.4), otherwise it SHOULD be set to 4 but MAY be set to any value from 1 to 7 to indicate a non-default priority. The values 8-15 are reserved for future use.

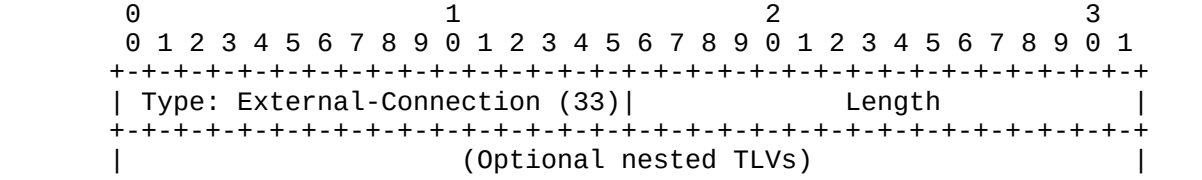
H-capability: Priority value used for electing the on-link DHCPv6 server offering non-temporary addresses. It MUST be set to 0 if the router is not capable of providing such addresses, otherwise it SHOULD be set to 4 but MAY be set to any value from 1 to 7 to indicate a non-default priority. The values 8-15 are reserved for future use.

L-capability: Priority value used for electing the on-link DHCPv4 server. It MUST be set to 0 if the router is not capable of running a legacy DHCPv4 server offering IPv4 addresses to clients, otherwise it SHOULD be set to 4 but MAY be set to any value from 1

to 7 to indicate a non-default priority. The values 8-15 are reserved for future use.

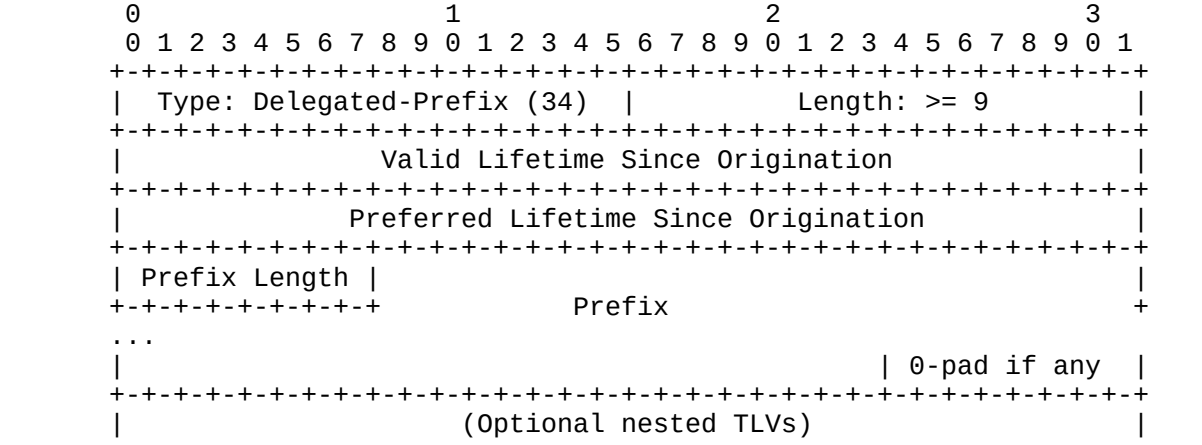
User-Agent: The user-agent is a human-readable UTF-8 string that describes the name and version of the current HNCP implementation.

10.2. External-Connection TLV



An External-Connection TLV is a container TLV used to gather network configuration information associated with a single external connection (Section 6.2) to be shared across the HNCP network. A node MAY publish an arbitrary number of instances of this TLV to share the desired number of external connections. Upon reception, the information transmitted in any nested TLVs is used for the purposes of prefix assignment (Section 6.3) and host configuration (Section 7).

10.2.1. Delegated-Prefix TLV



The Delegated-Prefix TLV is used by HNCP routers to advertise prefixes that are allocated to the whole network and can be used for prefix assignment. Delegated-Prefix TLVs are only valid inside External-Connection TLVs, and their prefixes MUST NOT overlap with those of other such TLVs in the same container.

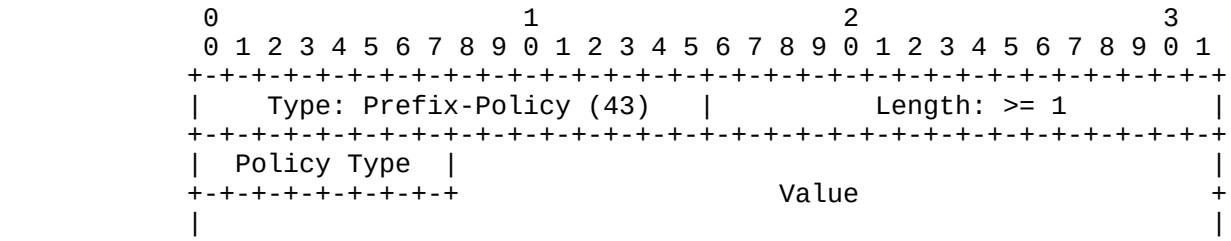
Valid Lifetime Since Origination: The time in seconds the delegated prefix was valid for at the origination time of the node data containing this TLV. The value MUST be updated whenever the node republishes its Node-State TLV.

Preferred Lifetime Since Origination: The time in seconds the delegated prefix was preferred for at the origination time of the node data containing this TLV. The value MUST be updated whenever the node republishes its Node-State TLV.

Prefix Length: The number of significant bits in the prefix.

Prefix: Significant bits of the prefix padded with zeros up to the next byte boundary.

10.2.1.1. Prefix-Policy TLV



The Prefix-Policy TLV contains information about the policy or applicability of a delegated prefix. This information can be used to determine whether prefixes for a certain use case (e.g., local reachability, Internet connectivity) do exist or are to be acquired and to make decisions about assigning prefixes to certain links or to fine-tune border firewalls. See Section 6.2 for a more in-depth discussion. This TLV is only valid inside a Delegated-Prefix TLV.

Policy Type: The type of the policy identifier.

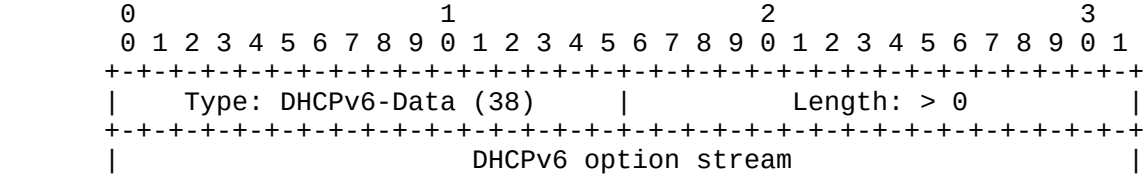
- 0: Internet connectivity (no value).
- 1-128: Explicit destination prefix with the Policy Type being the actual length of the prefix and the value containing significant bits of the destination prefix padded with zeros up to the next byte boundary.
- 129: DNS domain. The value contains a DNS label sequence encoded per [RFC1035]. Compression MUST NOT be used. The label sequence MUST end with an empty label.
- 130: Opaque UTF-8 string (e.g., for administrative purposes).

131: Restrictive assignment (no value).

132-255: Reserved for future additions.

Value: A variable-length identifier of the given type.

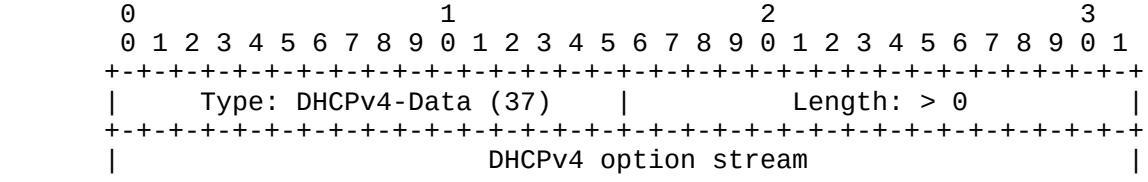
10.2.2. DHCPv6-Data TLV



This TLV is used to encode auxiliary IPv6 configuration information (e.g., recursive DNS servers) encoded as a stream of DHCPv6 options. It is only valid in an External-Connection TLV or a Delegated-Prefix TLV encoding an IPv6 prefix and MUST NOT occur more than once in any single container. When included in an External-Connection TLV, it contains DHCPv6 options relevant to the external connection as a whole. When included in a delegated prefix, it contains options mandatory to handle said prefix.

DHCPv6 option stream: DHCPv6 options encoded as specified in [RFC3315].

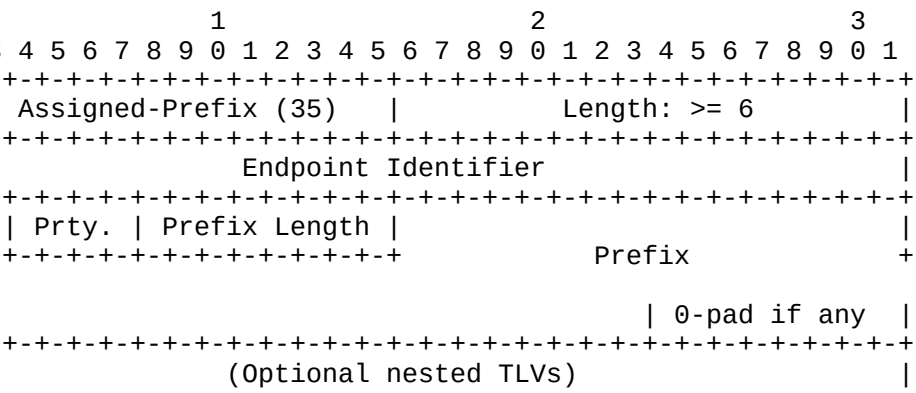
10.2.3. DHCPv4-Data TLV



This TLV is used to encode auxiliary IPv4 configuration information (e.g., recursive DNS servers) encoded as a stream of DHCPv4 options. It is only valid in an External-Connection TLV and MUST NOT occur more than once in any single container. It contains DHCPv4 options relevant to the external connection as a whole.

DHCPv4 option stream: DHCPv4 options encoded as specified in [RFC2131].

10.3. Assigned-Prefix TLV



This TLV is used to announce Published Assigned Prefixes for the purposes of prefix assignment (Section 6.3).

Endpoint Identifier: The endpoint identifier of the local interface the prefix is assigned to, or 0 if it is assigned to a Private Link (e.g., when the prefix is assigned for downstream prefix delegation).

Rsv.: Bits are reserved for future use. They MUST be set to 0 when creating this TLV, and their value MUST be ignored when processing the TLV.

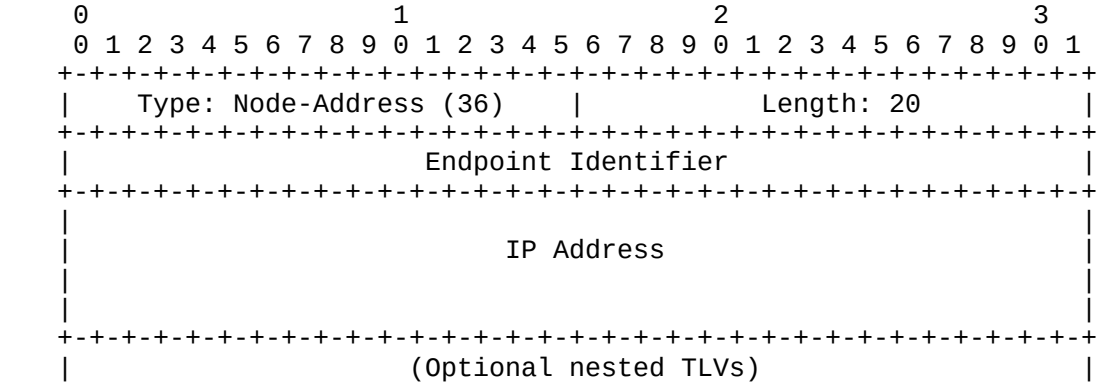
Prty: The Advertised Prefix Priority from 0 to 15.

- 0-1: Low priorities.
- 2: Default priority.
- 3-7: High priorities.
- 8-11: Administrative priorities. MUST NOT be used unless configured otherwise.
- 12-14: Reserved for future use.
- 15: Provider priorities. MAY only be used by the router advertising the corresponding delegated prefix and based on static or dynamic configuration (e.g., for excluding a prefix based on the DHCPv6-PD Prefix Exclude Option [RFC6603]).

Prefix Length: The number of significant bits in the Prefix field.

Prefix: The significant bits of the prefix padded with zeros up to the next byte boundary.

10.4. Node-Address TLV

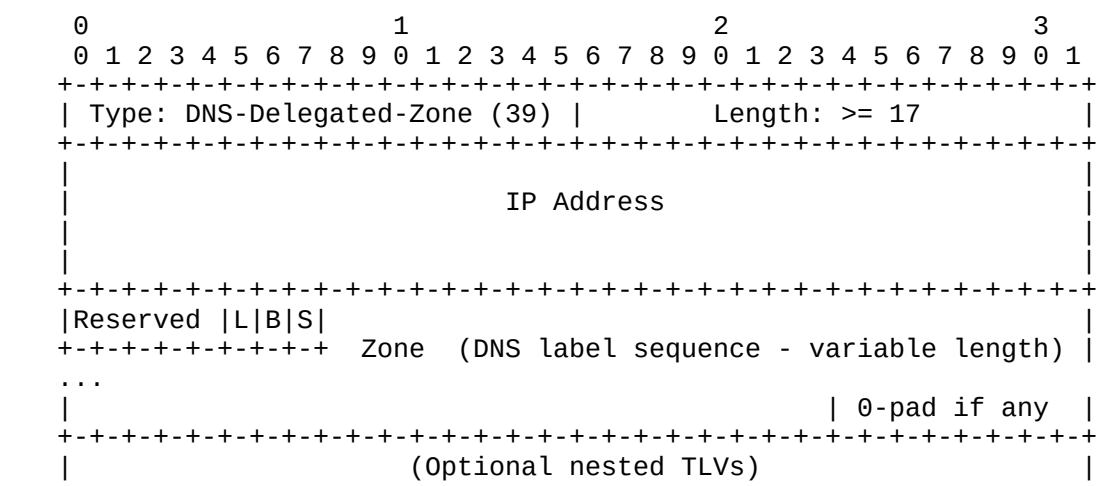


This TLV is used to announce addresses assigned to an HNCP node as described in Section 6.4.

Endpoint Identifier: The endpoint identifier of the local interface the prefix is assigned to, or 0 if it is not assigned on an HNCP enabled link.

IP Address: The globally scoped IPv6 address, or the IPv4 address encoded as an IPv4-mapped IPv6 address [RFC4291].

10.5. DNS-Delegated-Zone TLV



This TLV is used to announce a forward or reverse DNS zone delegation in the HNCP network. Its meaning is roughly equivalent to specifying an NS and A/AAAA record for said zone. Details are specified in Section 8.

IP Address: The IPv6 address of the authoritative DNS server for the zone; IPv4 addresses are represented as IPv4-mapped addresses [RFC4291]. The special value of :: (all zeros) means the delegation is available in the global DNS hierarchy.

Reserved: Those bits MUST be set to 0 when creating the TLV and ignored when parsing it unless defined in a later specification.

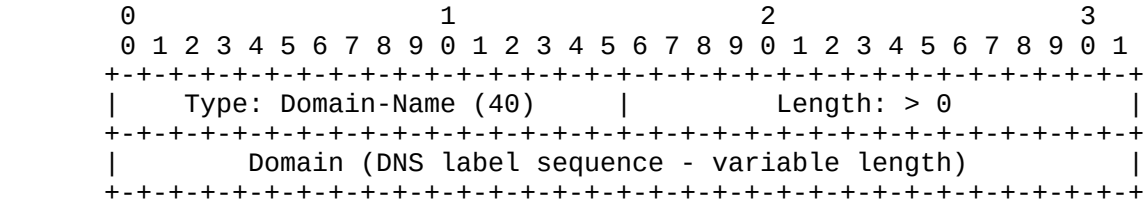
L-bit: (DNS-based Service Discovery (DNS-SD) [RFC6763] Legacy-Browse) indicates that this delegated zone SHOULD be included in the network's DNS-SD legacy browse list of domains at lb._dns-sd._udp.(DOMAIN-NAME). Local forward zones SHOULD have this bit set; reverse zones SHOULD NOT.

B-bit: (DNS-SD [RFC6763] Browse) indicates that this delegated zone SHOULD be included in the network's DNS-SD browse list of domains at b._dns-sd._udp.(DOMAIN-NAME). Local forward zones SHOULD have this bit set; reverse zones SHOULD NOT.

S-bit: (Fully qualified DNS-SD [RFC6763] domain) indicates that this delegated zone consists of a fully qualified DNS-SD domain, which should be used as the base for DNS-SD domain enumeration, i.e., _dns-sd._udp.(Zone) exists. Forward zones MAY have this bit set; reverse zones MUST NOT. This can be used to provision a DNS search path to hosts for non-local services (such as those provided by an ISP or other manually configured service providers). Zones with this flag SHOULD be added to the search domains advertised to clients.

Zone: The label sequence encoded according to [RFC1035]. Compression MUST NOT be used. The label sequence MUST end with an empty label.

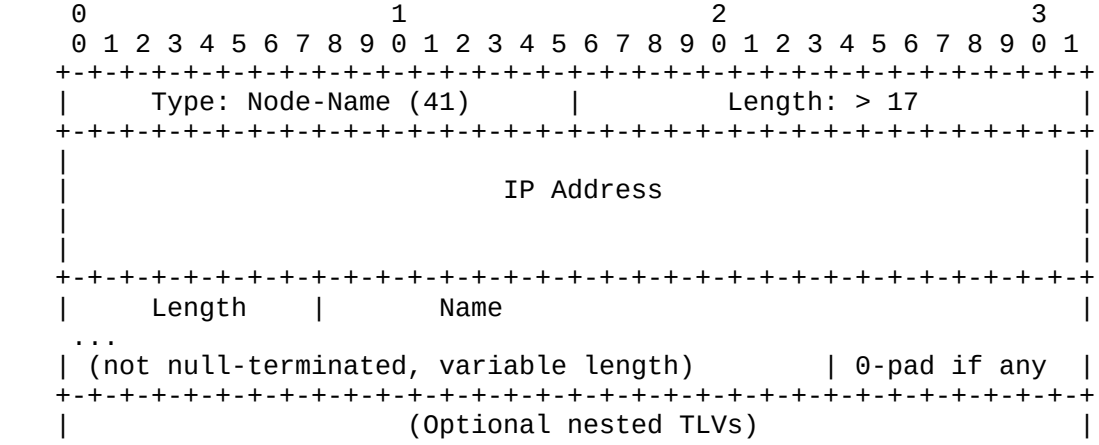
10.6. Domain-Name TLV



This TLV is used to indicate the base domain name for the network as specified in Section 8. This TLV MUST NOT be announced unless the domain name was explicitly configured by an administrator.

Domain: The label sequence encoded according to [RFC1035]. Compression MUST NOT be used. The label sequence MUST end with an empty label.

10.7. Node-Name TLV



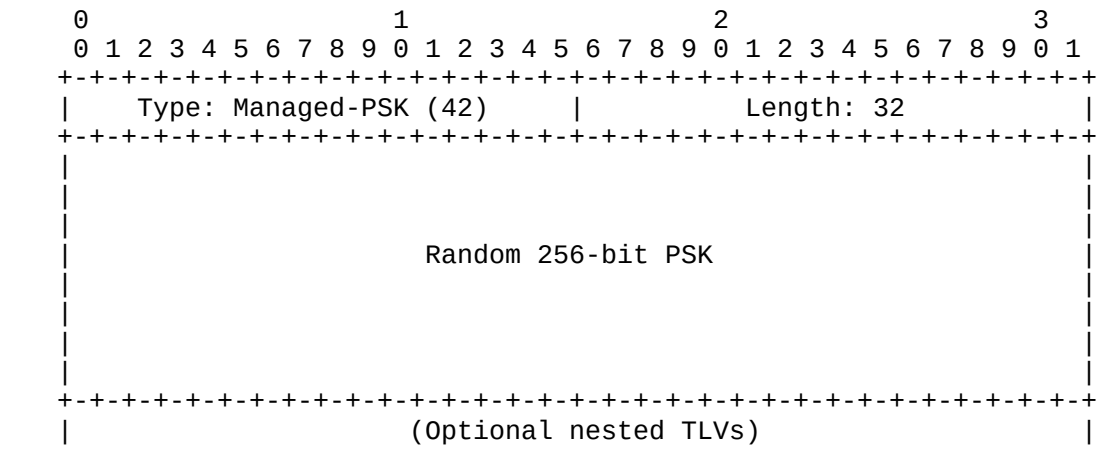
This TLV is used to assign the name of a node in the network to a certain IP address as specified in Section 8.

IP Address: The IP address associated with the name. IPv4 addresses are encoded using IPv4-mapped IPv6 addresses.

Length: The length of the name (0-63).

Name: The name of the node as a single DNS label.

10.8. Managed-PSK TLV



This TLV is used to announce a PSK for securing third-party protocols exclusively supporting symmetric cryptography as specified in Section 9.

11. General Requirements for HNCP Nodes

Each node implementing HNCP is subject to the following requirements:

- It MUST implement HNCP versioning (Section 4) and interface classification (Section 5).
- It MUST implement and run the method for securing third-party protocols (Section 9) whenever it uses the security mechanism of HNCP.

If the node is acting as a router, then the following requirements apply in addition:

- It MUST support Autonomous Address Configuration (Section 6) and configuration of hosts and non-HNCP routers (Section 7).
- It SHOULD implement support for naming and service discovery (Section 8) as defined in this document.
- It MAY be able to provide connectivity to IPv4 devices using DHCPv4.
- It SHOULD be able to delegate prefixes to legacy IPv6 routers using DHCPv6-PD (Section 6.3.4).

- In addition, the normative language of "Basic Requirements for IPv6 Customer Edge Routers" [RFC7084] applies with the following adjustments:
  - The generic requirements G-4 and G-5 are relaxed such that any known default router on any interface is sufficient for a router to announce itself as the default router; similarly, only the loss of all such default routers results in self-invalidation.
  - "WAN-Side Configuration" (Section 4.2) applies to interfaces classified as external.
  - If the Customer Edge (CE) sends a size hint as indicated in WPD-2, the hint MUST NOT be determined by the number of LAN interfaces of the CE but SHOULD instead be large enough to at least accommodate prefix assignments announced for existing delegated or ULA prefixes, if such prefixes exist and unless explicitly configured otherwise.
  - The dropping of packets with a destination address belonging to a delegated prefix mandated in WPD-5 MUST NOT be applied to destinations that are part of any prefix announced using an Assigned-Prefix TLV by any HNCP router in the network.
  - "LAN-Side Configuration" (Section 4.3) applies to interfaces not classified as external.
  - The requirement L-2 to assign a separate /64 to each LAN interface is replaced by the participation in the prefix assignment mechanism (Section 6.3) for each such interface.
  - The requirement L-9 is modified, in that the M flag MUST be set if and only if a router connected to the respective Common Link is advertising a non-zero H-capability. The O flag SHOULD always be set.
  - The requirement L-12 to make DHCPv6 options available is adapted, in that Canonical Encoding Rules (CER) SHOULD publish the subset of options using the DHCPv6-Data TLV in an External-Connection TLV. Similarly, it SHOULD do the same for DHCPv4 options in a DHCPv4-Data TLV. DHCPv6 options received inside an OPTION_IAPREFIX [RFC3633] MUST be published using a DHCPv6-Data TLV inside the respective Delegated-Prefix TLV. HNCP routers SHOULD make relevant DHCPv6 and DHCPv4 options available to clients, i.e., options contained in External-Connection TLVs that also include delegated prefixes from which a subset is assigned to the respective link.

* The requirement L-13 to deprecate prefixes is applied to all delegated prefixes in the network from which assignments have been made on the respective interface. Furthermore, the Prefix Information Options indicating deprecation MUST be included in Router Advertisements for the remainder of the prefixes' respective valid lifetime but MAY be omitted after at least 2 hours have passed.

12. Security Considerations

HNCP enables self-configuring networks, requiring as little user intervention as possible. However, this zero-configuration goal usually conflicts with security goals and introduces a number of threats.

General security issues for existing home networks are discussed in [RFC7368]. The protocols used to set up addresses and routes in such networks to this day rarely have security enabled within the configuration protocol itself. However, these issues are out of scope for the security of HNCP itself.

HNCP is a DNCP-based state synchronization mechanism carrying information with varying threat potential. For this consideration, the payloads defined in DNCP and this document are reviewed:

- Network topology information such as HNCP nodes and their common links.
- Address assignment information such as delegated and assigned prefixes for individual links.
- Naming and service discovery information such as auto-generated or customized names for individual links and nodes.

12.1. Interface Classification

As described in Section 5.3, an HNCP node determines the internal or external state on a per-interface basis. A firewall perimeter is set up for the external interfaces, and for internal interfaces, HNCP traffic is allowed, with the exception of the Leaf and Guest subcategories.

Threats concerning automatic interface classification cannot be mitigated by encrypting or authenticating HNCP traffic itself since external routers do not participate in the protocol and often cannot be authenticated by other means. These threats include propagation of forged uplinks in the homenet in order to, e.g., redirect traffic

destined to external locations and forged internal status by external routers to, e.g., circumvent the perimeter firewall.

It is therefore imperative to either secure individual links on the physical or link layer or preconfigure the adjacent interfaces of HNCP routers to an appropriate fixed category in order to secure the homenet border. Depending on the security of the external link, eavesdropping, man-in-the-middle, and similar attacks on external traffic can still happen between a homenet border router and the ISP; however, these cannot be mitigated from inside the homenet. For example, DHCPv4 has defined [RFC3118] to authenticate DHCPv4 messages, but this is very rarely implemented in large or small networks. Further, while PPP can provide secure authentication of both sides of a point-to-point link, it is most often deployed with one-way authentication of the subscriber to the ISP, not the ISP to the subscriber.

12.2. Security of Unicast Traffic

Once the homenet border has been established, there are several ways to secure HNCP against internal threats like manipulation or eavesdropping by compromised devices on a link that is enabled for HNCP traffic. If left unsecured, attackers may perform arbitrary traffic redirection, eavesdropping, spoofing, or denial-of-service attacks on HNCP services such as address assignment or service discovery, and the protocols are secured using HNCP-derived keys such as routing protocols.

Detailed interface categories like "Leaf" or "Guest" can be used to integrate not fully trusted devices to various degrees into the homenet by not exposing them to HNCP traffic or by using firewall rules to prevent them from reaching homenet-internal resources.

On links where this is not practical and lower layers do not provide adequate protection from attackers, DTLS-based secure unicast transport MUST be used to secure traffic.

12.3. Other Protocols in the Home

IGPs and other protocols are usually run alongside HNCP; therefore, the individual security aspects of the respective protocols must be considered. It can, however, be summarized that many protocols to be run in the home (like IGPs) provide -- to a certain extent -- similar security mechanisms. Most of these protocols do not support encryption and only support authentication based on Pre-Shared Keys natively. This influences the effectiveness of any encryption-based security mechanism deployed by HNCP as homenet routing information is thus usually not encrypted.

13. IANA Considerations

IANA has set up a registry for the (decimal values within range 32-511) "HNCP TLV Types" under "Distributed Node Consensus Protocol (DNCP)". The registration procedures is 'RFC Required' [RFC5226]. The initial contents are:

32: HNCP-Version

33: External-Connection

34: Delegated-Prefix

35: Assigned-Prefix

36: Node-Address

37: DHCPv4-Data

38: DHCPv6-Data

39: DNS-Delegated-Zone

40: Domain-Name

41: Node-Name

42: Managed-PSK

43: Prefix-Policy

44-511: Unassigned.

768-1023: Reserved for Private Use. This range is used by HNCP for per-implementation experimentation. How collisions are avoided is outside the scope of this document.

IANA has registered the UDP port numbers 8231 (service name: hncp-udp-port, description: HNCP) and 8232 (service name: hncp-dtls-port, description: HNCP over DTLS), as well as an IPv6 link-local multicast address FF02:0:0:0:0:0:0:11 (description: All-Homenet-Nodes).

14. References

14.1. Normative References

[RFC1321] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, April 1992.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.

[RFC2132] Alexander, S. and R. Droms, "DHCP Options and BOOTP Vendor Extensions", RFC 2132, March 1997.

[RFC3004] Stump, G., Droms, R., Gu, Y., Vyaghrapuri, R., Demirtjis, A., Beser, B., and J. Privat, "The User Class Option for DHCP", RFC 3004, November 2000.

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

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

[RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, October 2005.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, DOI 10.17487/RFC5226, May 2008, <http://www.rfc-editor.org/info/rfc5226>.

[RFC6092] Woodyatt, J., Ed., "Recommended Simple Security Capabilities in Customer Premises Equipment (CPE) for Providing Residential IPv6 Internet Service", RFC 6092, DOI 10.17487/RFC6092, January 2011, <http://www.rfc-editor.org/info/rfc6092>.

[RFC6206] Levis, P., Clausen, T., Hui, J., Gnawali, O., and J. Ko, "The Trickle Algorithm", RFC 6206, March 2011.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, January 2012.

[RFC6603] Korhonen, J., Savolainen, T., Krishnan, S., and O. Troan, "Prefix Exclude Option for DHCPv6-based Prefix Delegation", RFC 6603, May 2012.

[RFC6763] Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", RFC 6763, February 2013.

[RFC7217] Gont, F., "A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC)", RFC 7217, DOI 10.17487/RFC7217, April 2014, <http://www.rfc-editor.org/info/rfc7217>.

[RFC7695] Pfister, P., Paterson, B., and J. Arkko, "Distributed Prefix Assignment Algorithm", RFC 7695, DOI 10.17487/RFC7695, November 2015, <http://www.rfc-editor.org/info/rfc7695>.

[RFC7787] Stenberg, M. and S. Barth, "Distributed Node Consensus Protocol", RFC 7787, DOI 10.17487/RFC7787, February 2016, <http://www.rfc-editor.org/info/rfc7787>.

14.2. Informative References

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

[RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.

[RFC3118] Droms, R. and W. Arbaugh, "Authentication for DHCP Messages", RFC 3118, June 2001.

[RFC6234] Eastlake, D. and T. Hansen, "US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)", RFC 6234, May 2011.

[RFC6241] Enns, R., Bjorklund, M., Schoenwaelder, J., and A. Bierman, "Network Configuration Protocol (NETCONF)", RFC 6241, June 2011.

[RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, February 2013.

[RFC7084] Singh, H., Beebee, W., Donley, C., and B. Stark, "Basic Requirements for IPv6 Customer Edge Routers", RFC 7084, November 2013.

[RFC7368] Chown, T., Arkko, J., Brandt, A., Troan, O., and J. Weil, "IPv6 Home Networking Architecture Principles", RFC 7368, October 2014.

[RFC7525] Sheffer, Y., Holz, R., and P. Saint-Andre, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195, RFC 7525, DOI 10.17487/RFC7525, May 2015, <http://www.rfc-editor.org/info/rfc7525>.

Appendix A. Acknowledgments

Thanks to Ole Troan, Mark Baugher, Mark Townsley, Juliusz Chroboczek, and Thomas Clausen for their contributions to the document.

Thanks to Eric Kline for the original border discovery work.

Authors' Addresses

Markus Stenberg
Independent
Helsinki 00930
Finland

Email: markus.stenberg@iki.fi

Steven Barth
Independent
Halle 06114
Germany

Email: stbarth@cisco.com

Pierre Pfister
Cisco Systems
Paris
France

Email: pierre.pfister@darou.fr

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

Table of Contents

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:

- 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).

- 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).

- 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.

- 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.

- 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-hncp] Stenberg, M., Barth, S., and P. Pfister, "Home Networking Control Protocol", draft-ietf-homenet-hncp-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>.

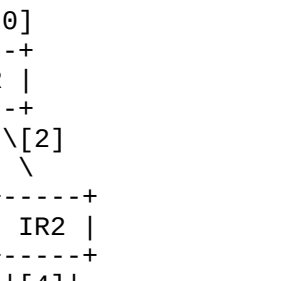
9.3. URIs

[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:

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

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

- 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: April 21, 2016

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

R. Weber
Nominum
W. Cloetens
SoftAtHome

October 19, 2015

DHCPv6 Options for Homenet Naming Architecture
draft-ietf-homenet-naming-architecture-dhc-options-03.txt

Abstract

Customer Premises Equipment (CPE) devices are usually constrained devices with reduced network and CPU capabilities. As such, a CPE exposing the authoritative naming service for its home network on the Internet may become vulnerable to resource exhaustion attacks. One way to avoid exposing CPE is to outsource the authoritative service to a third party, e.g. ISP. Such an outsource requires setting up an architecture which may be inappropriate for most end users. This document defines DHCPv6 options so any agnostic CPE 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 April 21, 2016.

Table of Contents

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 CPE. This key is used as an authentication credential for the CPE.

- 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

CPEs are usually constrained devices with reduced network and CPU capacities. As such, a CPE 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 CPE 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 CPE 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 CPE 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 CPE 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 CPE. 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 CPE 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 CPE, and only the CPE should be able to update or upload these zones. To authenticate the CPE, this document defines the Client Public Key Option. This option is sent by the CPE to the DHCPv6 server and provides the Client Public Key the CPE 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 CPE 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 thrustworhy connection between the CPE and the DHCPv6 server is defined in [I-D.ietf-dhc-sedhcpv6]. When both links are trusted, the CPE 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 CPE 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 CPE needs to interact with the server, then, the end user is expected to provide the CPE’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 CPE 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 CPE 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 CPE may behave and be configured regarding various scenarios.

4. Protocol Overview

This section provides an overview of the CPE'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 CPE 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 CPE 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 CPE. Once each server got the public key, the CPE 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 CPE 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 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 CPE 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 CPE 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 CPE 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 CPE has retrieved the DNS Zone Template, the CPE can build the Homenet Zone and the Homenet Reverse Zone. Eventually the CPE signs these zones.

- 5: Once the Homenet Reverse Zone has been set, the CPE 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 CPE 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.

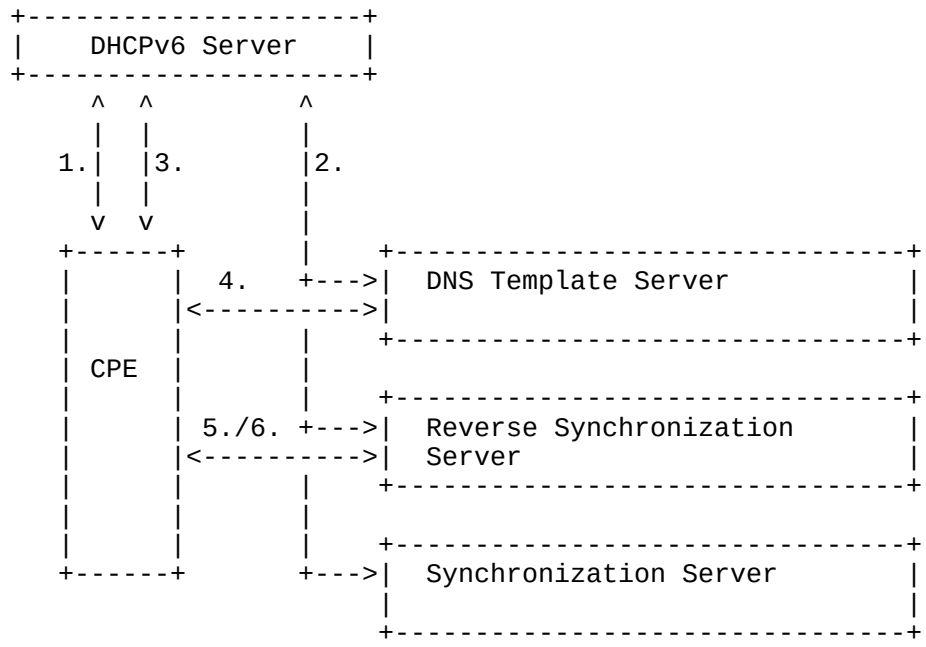


Figure 1: Protocol Overview

As described above, the CPE is likely to interact with various DNS content. More specifically, the CPE 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 CPE 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 CPE, 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 CPE 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 CPE. The appropriate authentication methods may, for example, be chosen according to the level of confidentiality or the level of authentication requested by the CPE 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 CPE 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 CPE 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 CPE 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 CPE 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 CPE should proceed to the DNSSEC signature checks. If signature check fails, it MUST reject the response. If the signature check succeeds, the CPE 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. CPE Configuration

5.1. CPE Primary / Secondary Synchronization Configurations

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

When the CPE is plugged its IP address may be unknown to the secondary. The section details how the CPE 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. CPE / Synchronization Server

The CPE 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 CPE. The only missing information is the IP address of the CPE. This IP address is provided by the CPE by sending a NOTIFY [RFC1996].

When the CPE 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. CPE / Reverse Synchronization Server

The CPE 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. CPE 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 CPE 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 CPE. 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 CPE. 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 CPE 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 CPE 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 CPE 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 CPE in order to perform a transaction with the DNS server. See Section 4.2 for more details.

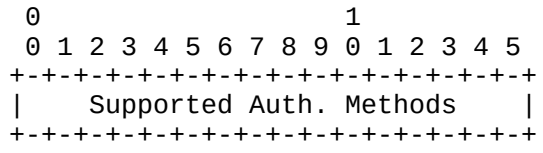


Figure 2: Supported Authentication Methods Filed

- 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.

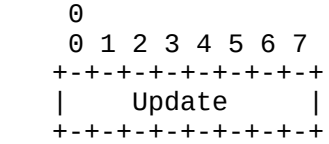


Figure 3: Update Field

- 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 CPE. 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 CPE how to retrieve the Homenet Zone Template. It provides a FQDN the CPE 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.

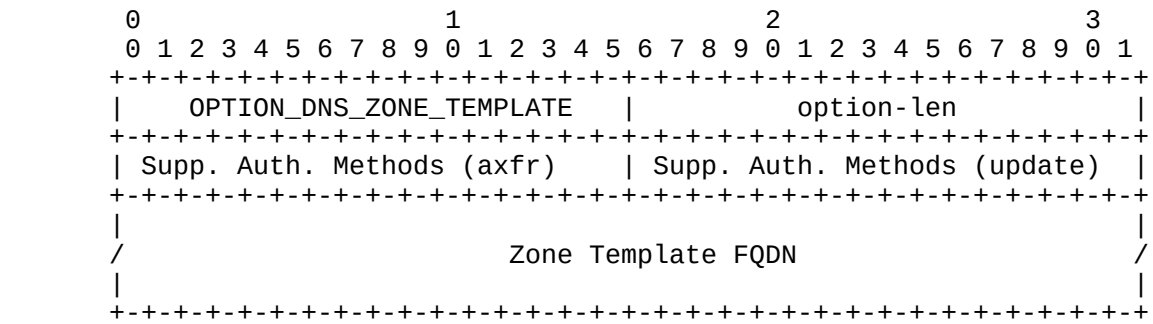


Figure 4: Zone Template Option

- 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 CPE 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.

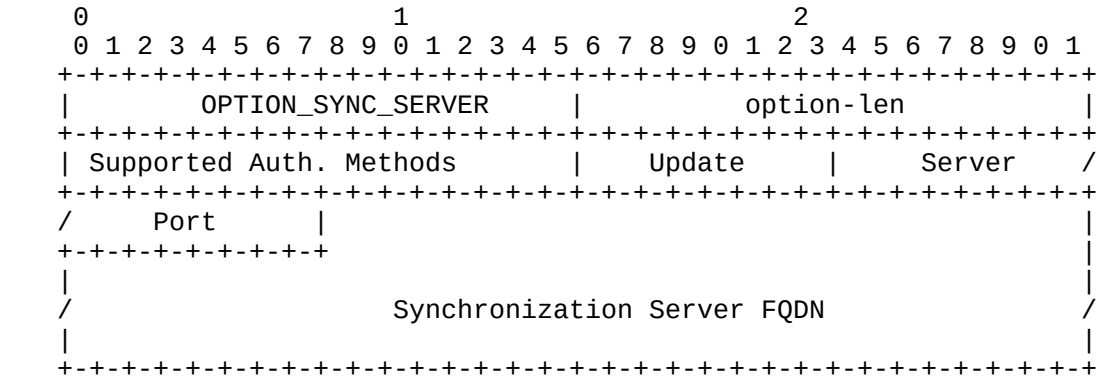


Figure 5: Synchronization Server Option

- 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 CPE 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.

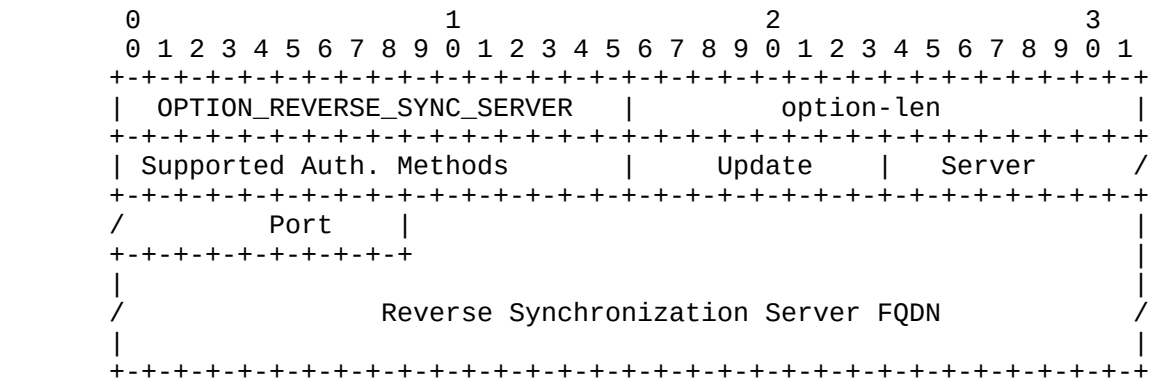


Figure 6: Reverse Synchronization Server Option

- 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 CPE. 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 CPE 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 CPE 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 CPE.

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 CPE 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 CPE or by a third party. We recommend the zone to be signed by the CPE, and that the signed zone is uploaded.

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

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

The document considers that the channel between the CPE and the ISP DHCP Server is trusted. More specifically, the CPE 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. CPEs are sensitive to DoS

CPE have not been designed for handling heavy load. The CPE 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., Shen, S., Zhang, D., and T. Jinmei, "Secure DHCPv6", draft-ietf-dhc-sedhcpv6-08 (work in progress), June 2015.

[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-dnsext-cname-dname]
Sury, O., "CNAME+DNAME Name Redirection", draft-sury-dnsext-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 CPE 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 CPE to enable secure authenticated transaction between the CPE and these DNS servers. More specifically, credentials are provided to:

- Synchronization Server
- Reverse Synchronization Server
- DNS Template Server

The CPE 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-dnsext-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 CPE 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 CPE. Once this modification has been performed, the CPE can be changed, the Client Public Key of the CPE may be changed, this does not need to be done another time. One can imagine a GUI on the CPE 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 CPE, that is the Client Public Key of the CPE, 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 CPE, this operation is likely to be performed every time the CPE is changed, and every time the Client Public Key generated by the CPE 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 CPE need to be configured for every CPE. Configuration is expected to be CPE live-long.

A.4. Multiple ISPs

This scenario considers a CPE connected to multiple ISPs.

Firstly, suppose the CPE has been configured with the based scenarios exposed in Appendix A.1. The CPE has multiple interfaces, one for each ISP, and each of these interface is configured using DHCP. The CPE 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 CPE Client Public Key may be shared between the CPE and the multiple ISPs. The CPE 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 CPE connected to multiple ISPs with multiple Registered Homenet Domains. However, the CPE 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 CPE 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 CPE 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 CPE 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 CPE. This latter configuration is likely to match most CPE implementations.

The protocol and DHCPv6 options described in this document are fully compatible with a CPE connected to multiple ISPs. To configure or not and how to configure the CPE depends on the CPE facilities. Appendix A.1 and Appendix A.2 require the CPE 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 CPE.

- 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 CPE 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

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

Email: tomasz.mrugalski@gmail.com
URI: http://www.isc.org

Chris Griffiths

Email: cgriffiths@gmail.com

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

Email: ralf.weber@nominum.com
URI: http://www.nominum.com

Wouter Cloetens
SoftAtHome
vaartdijk 3 701
3018 Wijgmaal
Belgium

Email: wouter.cloetens@softathome.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

Table of Contents

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:

- Provisioning of a domain name under which names can be published and services advertised
- Associating names that are subdomains of that name with hosts.
- Advertising services available on the local network by publishing resource records on those names.
- Distribution of names published in that namespace to servers that can be queried in order to resolve names
- Correct advertisement of name servers that can be queried in order to resolve names
- 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:bee7::/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:bee7::1 would therefore yield a key of '1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0'

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:

- The global name
- The ZSK for both local namespaces
- 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 ueries 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-robining 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 send 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 recevied, 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 received 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