Homenet Working Group
Internet-Draft
Intended status: Informational
Expires: September 10, 2015

S. Barth

March 9, 2015

Incremental Deployment of HNCP and IGPs in home networks
draft-barth-homenet-incremental-deployment-00

Abstract

This document describes an incremental approach towards deploying HNCP and routing protocols in home networks. Its aim is to provide a minimal, forward-compatible transitional extension to HNCP to promote testing, deployment and adoption of homenet technology while the IGP decision and standardization process is not yet finalized.

Status of This Memo

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

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 10, 2015.

Table of Contents

1. Introduction

While it is expected that the average number of routers and their hardware and software capabilities in a typical home network will grow over time, this trend has historically been gradual. Thus it can be expected that in the near future there will most likely be not more than a handful of routers in a home and their capabilities for operating in a fully-routed mode are limited.

It is also hard to predict which types of other networks the homenet technology will be used in and what attributes these networks will have (e.g. number of routers, links, link-types, topologies, suitable methods for detecting link-layer status and deriving metric). Furthermore the standardization of autoconfiguring and source-dest-routing capable protocols is not expected to be finalized soon and there is currently a shortage of widely testable or deployable implementations fulfilling these criteria.

These issues and a general lack of consensus over an all-purpose routing protocol support a transitional forward-compatible extension to HNCP [I-D.ietf-homenet-hncp] implementations in order to advance homenet progress and to promote adoption. This draft describes a solution sufficient for small networks allowing gradual adoption of homenet principles into existing networks while ensuring an easy transition to a future standardized version.

2. Implementation Requirements for Incremental Deployments

Each homenet router runs an incremental connectivity algorithm at all times on each network interface it is running HNCP on and can optionally additionally run one or more routing protocols.

A router running a routing protocol alongside the incremental connectivity algorithm must strictly prefer all routes of the routing protocol over all routes generated by the incremental connectivity algorithm, except for those having destinations not already known to the routing protocol but that lie within one of the designated

prefixes of the homenet (i.e. prefixes assigned by a homenet router that does not speak the respective routing protocol). In case of routers running more than one routing protocol alongside the incremental connectivity algorithm, all such routers in the homenet ensure that in doing so they do not cause routing loops, e.g. by agreeing upon a network-wide order in which routes of the protocols are considered.

3. Incremental Connectivity Algorithm

This algorithm is designed to provide connectivity in small home networks that would not benefit from exploiting link characteristics or metrics. It is intentionally kept simple to require no additional TLV-information by reusing existing topology and address assignment information provided by HNCP and only requires minimal implementation overhead.

Each homenet router traverses the HNCP neighbor graph using a breadth-first search starting with its own node's immediate neighbors. Neighbors of a node are traversed in ascending order of their node identifier. During traversal the router determines the path to each other router (R), the next-hop neighbor N(R) and the number of hops to it D(R). The router then creates routes based on the following rules:

Create a route	To	From	Via	Metric
For each Prefix (A) in an Assigned Prefix TLV of any Router (R)	A	any	N(R)	D(R)
For each IPv6 prefix (P) in a Delegated Prefix TLV of any Router (R)	::/0	P	N(R)	D(R)
For the first Router (R) traversed that announces an IPv4 Delegated Prefix TLV	0.0.0.0/0	any	N(R)	D(R)

This process is repeated every time the router detects a change in the neighbor graph or prefix assignment information in the network.

4. Security Considerations

The mechanism described in this document is based on HNCP, thus security considerations for this document are already covered by [I-D.ietf-homenet-hncp].

5. IANA Considerations

This document has no actions for IANA.

6. Normative references

[I-D.ietf-homenet-hncp]
Stenberg, M., Barth, S., and P. Pfister, "Home Networking Control Protocol", draft-ietf-homenet-hncp-04 (work in progress), March 2015.

Appendix A. Acknowledgements

Thanks to Pierre Pfister and Markus Stenberg for comments and suggestions.

Author's Address

Steven Barth
Halle 06114
Germany

Email: cyrus@openwrt.org

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 15, 2015

S. Cheshire
Apple Inc.

November 11, 2014

Special Use Top Level Domain "home"
draft-cheshire-homenet-dot-home-01

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 May 15, 2015.

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 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.

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 learn more about the current widespread use of ".home" names, and to document and formalize that usage.

Evidence [ICANN1][ICANN2] indicates that ".home" queries frequently leak out and reach the root name servers. We speculate that this is 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 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.

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 requests, 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 requests (as may be facilated 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 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, November 1987.

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

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

[RFC6761] Cheshire, S. and M. Krochmal, "Special-Use Domain Names", RFC 6761, February 2013.

7.2. Informative References

[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.

[RFC3397] Aboba, B. and S. Cheshire, "Dynamic Host Configuration Protocol (DHCP) Domain Search Option", RFC 3397, November 2002.

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

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

[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: Informational
Expires: August 16, 2015

C. Donley
M. Kloberdans
CableLabs
J. Brzozowski
Comcast
C. Grundemann
ISOC

February 12, 2015

Customer Edge Router Identification Option
draft-donley-dhc-cer-id-option-05

Abstract

Addressing mechanisms supporting DHCPv6 Prefix Delegation in home networks such as those described in CableLabs' eRouter specification and the HIPnet Internet-Draft require identification of the customer edge router (CER) as the demarcation between the customer network and the service provider network. This document reserves a DHCPv6 option to identify the CER.

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 August 16, 2015.

Copyright Notice

Table of Contents

1. Introduction

Some addressing mechanisms supporting DHCPv6 Prefix Delegation in home networks such as those described in [I-D.grundemann-homenet-hipnet] and [EROUTER] require identification of the customer edge router as the demarcation between the customer network and the service provider network. For prefix delegation purposes, it is desirable for other routers within the home to know which device is the CER so that the customer home network only requests a single prefix from the ISP DHCPv6 server, and efficiently distributes this prefix within the home. CER-ID is a 128-bit string that optionally represents an IPV6 address, or another arbitrary number. The CER-ID maybe treated as a hint to be used with border detection methods. This document reserves a DHCPv6 option to be used to identify the CER.

1.1. Requirements Language

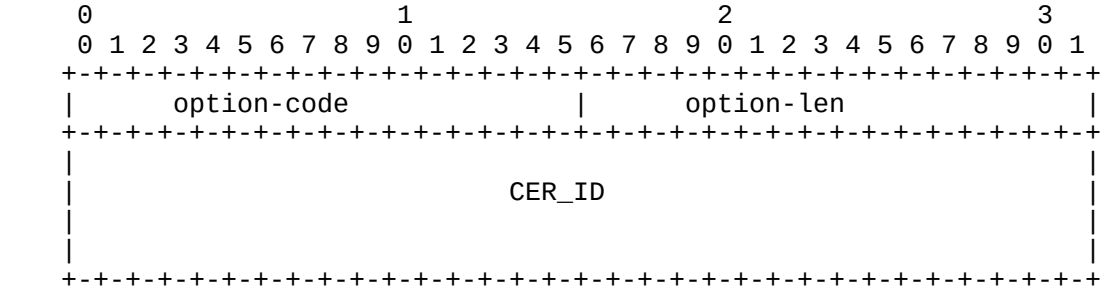
The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. CER Identification Option

A Customer Edge Router (CER) sets the CER_ID to the IPv6 address of its LAN interface. If it has more than one LAN IPv6 address, it selects one of its LAN or other non-WAN IPv6 addresses to be used as the CER_ID. An ISP server does not respond with the CER_ID or sets

the CER_ID to ::. Such a response or lack of response indicates to the DHCPv6 client that it is the CER.

The format of the CER Identification option is:



option-code OPTION_CER_ID (TBD).
option-len 36
CER_ID value IPv6 address of CER or ::

Figure 1.

A DHCPv6 client SHOULD include the CER Identification option code in an Option Request option [RFC3315] in its DHCP Solicit messages.

The DHCPv6 server MAY include the CER Identification option in any response it sends to a client that has included the CER Identification option code in an Option Request option. The CER Identification option is sent in the main body of the message to client, not as a sub-option in, e.g., an IA_NA, IA_TA [RFC3315]option.

When sending the CER Identification option, the DHCPv6 server MUST set the CER_ID value to either one of its IPv6 addresses, another identifier, or ::. If a device does not receive the CER Identification Option or receives a CER ID of :: from the DHCPv6 server, it MUST include one of its Globally Unique IPv6 addresses (unless another identifier is used), in the CER_ID value in response to DHCPv6 messages received by its DHCPv6 server that contains the CER Identification option code in an Option Request option. If the device has only one LAN interface, it SHOULD use its LAN IPv6 address as the CER_ID value. If the device has more than one LAN interface, it SHOULD use the lowest Globally Unique address.

3. CER-ID Compatibility

CER-ID explicitly indicates that a gateway is, or is not, the demarcation point between public and private networks by containing a reachable IPv6 address, other identifier or a double colon '::'

(double colon indicates that the CER-ID sender is NOT the edge router), and as a complement, can be applied to various border definitions and detection methods such as:

- I.D. Draft-IETF-Homenet-Arch-16 [I-D.ietf-homenet-arch]
- I.D. Draft-Grundemann-homenet-HIPnet-01 [I-D.grundemann-homenet-hipnet]
- I.D. Draft-IETF-Kline-Homenet-Default-Perimeter-01 [I-D.kline-default-perimeter]
- Others, including manual configuration

4. IANA Considerations

IANA is requested to assign an option code from the "DHCP Option Codes" Registry for OPTION_CER_ID. IANA is also requested to maintain a list of authentication options.

5. Security Considerations

The security of a home network is an important consideration. Both the HIPNet [I-D.grundemann-homenet-hipnet] and Homenet [I-D.ietf-homenet-arch] approaches change the operational model of the home network vs. today’s IPv4-only paradigm. Specifically, these networks eliminate NAT inside the home network (and only enable it for IPv4 at the edge router, if required), support global addressability of devices, and thus need to consider firewall and/or filter support in various home routers. As the security profile of these home routers can shift based on their position in the network (e.g., edge vs. internal), security can be severely compromised if routers misidentify their border and mistakenly reduce or eliminate firewall rules. If the CER-ID option is used as part of the border detection algorithm, it becomes a natural, but not the only place to enact firewall, NAT, Prefix Delegation and other functions in the home network. Further security is provided using the mechanisms defined in RFC 3315, DHCP for IPv6.

6. Acknowledgements

7. References

7.1. Normative References

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

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

7.2. Informative References

[EROUTER] CableLabs, "CableLabs IPv4 and IPv6 eRouter Specification (CM-SP-eRouter-I12-131120)", April 2014.

[I-D.grundemann-homenet-hipnet] Grundemann, C., Donley, C., Brzozowski, J., Howard, L., and V. Kuarsingh, "A Near Term Solution for Home IP Networking (HIPnet)", draft-grundemann-homenet-hipnet-01 (work in progress), February 2013.

[I-D.ietf-homenet-arch] Chown, T., Arkko, J., Brandt, A., Troan, O., and J. Weil, "IPv6 Home Networking Architecture Principles", draft-ietf-homenet-arch-16 (work in progress), June 2014.

[I-D.kline-default-perimeter] Kline, E., "Default Border Definition", draft-kline-default-perimeter-01 (work in progress), November 2012.

Authors' Addresses

Chris Donley
CableLabs
858 Coal Creek Cir.
Louisville, CO 80027
US

Email: c.donley@cablelabs.com

Michael Kloberdans
CableLabs
858 Coal Creek Cir
Louisville, CO 80027
US

Email: m.kloberdans@cablelabs.com

John Brzozowski
Comcast
1306 Goshen Parkway
West Chester, PA 19380
US

Email: john_brzozowski@cable.comcast.com

Chris Grundemann
ISOC
Denver CO

Email: cgrundemann@gmail.com

Homenet
Internet-Draft
Intended status: Standards Track
Expires: April 18, 2016

L. Geng
H. Deng
China Mobile

October 16, 2015

Use Cases for Multiple Provisioning Domain in Homenet
draft-geng-homenet-mpvd-use-cases-02

Abstract

This document describes the use cases of multiple provisioning domain (MPVD) in homenet. Although most residential networks nowadays are connected to a single ISP and normally subscribed to standard internet service, it is expected that much wider range of devices and services will become common in home networks. Homenet defines such home network topologies with increasing number of devices with the assumption that it requires minimum configuration by residential user. As described in the homenet architecture ([RFC7368]), multihoming and multi-service residential network will be more common in the near future. Nodes in such network may commonly have multiple interfaces or subscribe to multiple services. Potential types of PVD-aware nodes concerning interface and service specific provisioning domains are introduced in this document. Based on this, different MPVD configuration examples are given. These examples illustrate how PVD may be implemented in home network. PVDs provide independent provisioning domains for different interfaces and services, which enables robust and flexible network configuration for these networks.

Table of Contents

1. Introduction

It is believed that future residential network will more commonly be multihomed, which potentially provides either resilience or more flexible services. At the same time, more internal routing and multiple subnets are expected to commonly exist in such networks. For example, customer may want independent subnets for private and guest usages. Homenet describes such future home network involving multiple routers and subnets ([RFC7368]).

Multihoming and the increasing number of subnets bring challenges on provisioning of the network. As stated in [RFC6418], such multihomed scenarios with nodes attached to multiple upstream networks may experience configuration conflicts, leading to a number of problems. To deal with these problem, draft-ietf-mif-mpvd-arch-10 provides a framework which introduces Provisioning Domain (PvD), which associates a certain interface and its related network configuration information. Hence, corresponding network configuration can be used when packets are delivered through a particular interface.

This document focuses on the MPvD use cases in residential network, particularly the IPV6-based homenet. Based on the homenet topology, use cases of MPvD in homenet are described for both singlehomed and multihomed network configurations.

1.1. 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 RFC 2119 [RFC2119].

2. Terminology and Abbreviations

The terminology and abbreviations used in this document are defined in this section.

- ISP: Internet Service Provider. A traditional network operator who provides internet access to customers.

3. Homenet with Multiple PvDs

In the most common multihoming scenarios, the home network has multiple physical connections to the ISP networks. Section 3.2.2.2 and 3.2.2.3 in [RFC7368] give the topology examples of such homenet. In the examples, homenet hosts are connected to a single or multiple customer edge routers (CE router), the CE routers are then connected to separate ISP networks. For the particular topology with a single CE router given in Section 3.2.2.3 in [RFC7368], the CE router is a mif node since it has two interfaces connected to individual service provider routers. Given that the CE router is a PvD-aware node, it may have two PvDs provided by ISPs respectively.

Apart from the multihoming resulted from physical connections , PvDs in Homenet can also be used for service provisioning. For example, a host may subscribe one ISP for internet service, whilst subscribe to another ISP for Internet of Things (IoT) service given that the CE router have access to both ISPs. On the other hand, the host user may also subscribe to the same ISP for both services. In either

case, PvDs can be used for customized network configurations purposes. This enables the service providers to provide independent and flexible provisioning for different services. Meanwhile, IoT service providers may also want to use independent PvDs to avoid the configuration conflicts between each other as stated in RFC6418.

A typical example of a PvD in home network is the one associating corresponding network configuration with an HNCP routers. These includes both CE router and interior router in Homenet. As described in ([RFC7368]), an CE router in homenet may have one or more external interfaces with ISPs and internal interfaces with interior routers. For external interfaces, the CE router can receive associated PvD information from corresponding ISPs. For interior interfaces, the interior router can receive PvD information from connected CE router or other interior routers.

Hosts in homenet are expected to be multihomed as well. Hence, PvD may also be used in such cases to associate different network configurations. In this case, the PvD information is received from the HNCP router a host is attached to, either a CE router or a interior router.

4. Examples of MPvD Configurations in Home Network

This section gives some examples of MPvD configurations in home network.

4.1. Single CE Router Connected to Single ISP with interior router

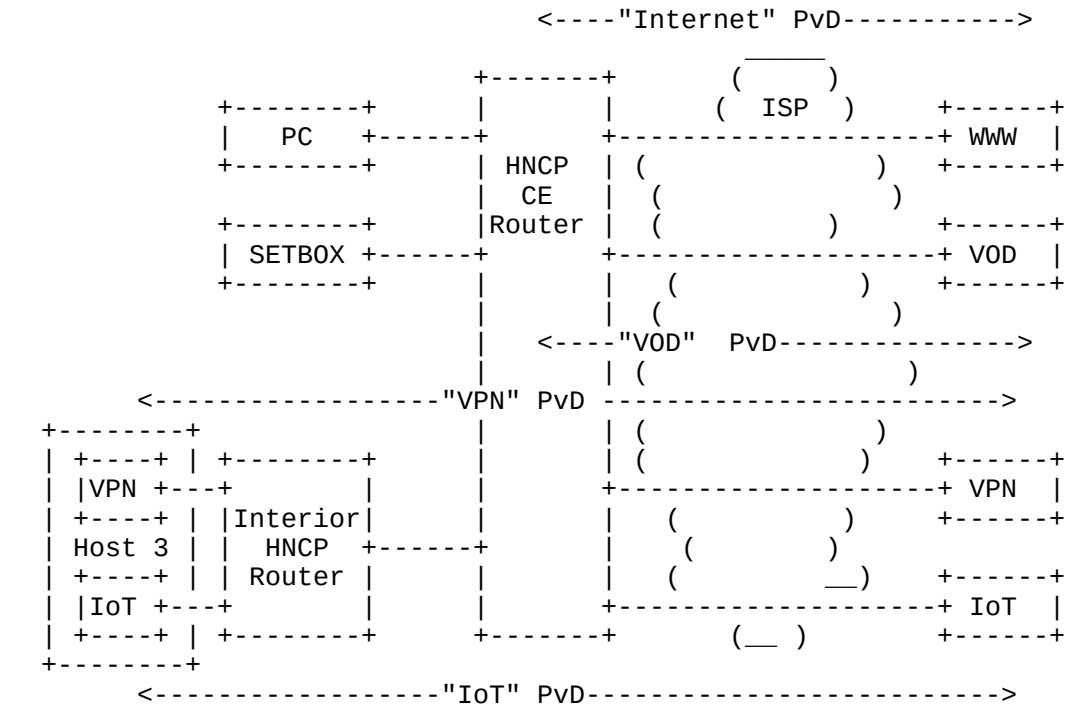


Figure 1

In this example, A homenet is connected with a single ISP as seen in Figure 1. Four different services are provided to this home network including Internet, VOD, VPN and IoT. There are 3 Hosts in this set-up. PC and setbox use "Internet" and "VoD" PvDs respectively and Host 3 uses both "VPN" and "IoT" PvD.

The four PVDs could be either implicit or explicit PvDs. Explicit PvDs should be initially assigned to HNCP CE router by ISP. And then forwarded to interior routers and host. Given that all PvDs are explicit in the case above, the "Internet" PvD is forwarded to PC and "VOD" PvD to SETBOX by the CE router, and the "VPN" and "IoT" PvDs are forwarded to interior HNCP router and then Host 3. The PvD_ID should be kept the same when the PvDs are forwarded. However, other associated network configuration (i.e. delegated prefixes) should be changed accordingly.

4.2. Two CE Routers Connected to two ISPs with Shared Subnets

To be added

4.3. One CE Routers Connected to two ISPs with Shared Subnets

To be added

5. PvD-aware node in homenet

As defined in MIF, "PvD-aware node is a a node that supports the association of network configuration information into PvDs and the use of these PvDs to serve requests for network connections". In Homenet, the HNCP CE router, interior router and host are all PvD-aware nodes.

The HNCP CE router PvD-related functionality is define as follows:

- Generates implicit PvDs for different uplink interfaces.
- Requests and receives all explicit PvDs provided by the connected ISPs.
- Generates explicit PvDs for interior routers and hosts referring to the ISP-provided PvDs and forwards them accordingly.
- Creates and stores the PvD mapping between the PvD applied itself the the one forwarded to interior routers and hosts using the assigned PvD_ID and prefix.
- Identify the prefix received from homenet nodes and performs PvD selection based on PvD mapping.

The interior router PvD-related functionality is defined as follows:

- Generates implicit PvDs for different homenet internal interface.
- Requests and receives all explicit PvDs provided by connected homenet routers.
- Generates explicit PvDs for interior routers and hosts referring to the homenet-router-provided PvDs and forwards them accordingly.
- Creates and stores the PvD mapping between the PvD applied itself the the one forwarded to interior routers and hosts using the assigned PvD_ID and prefix.

- o Identify the prefix received from homenet nodes and performs PvD selection based on PvD mapping.

The host PvD-related functionality is defined as follows:

- o Generates implicit PvDs for different interfaces between host and homenet routers.

- o Requests and receives all explicit PvDs provided by connected homenet routers.

6. IPv4 compatibility

PvD in homenet can be either single-family or dual-stack. For single-family PvD, the IPV4 and IPV6 configurations should be managed in separate PvDs with different PvD identities. For Dual-stack PvDs, IPV4 and IPV6 configurations can exist in the same PvD. In both cases, there can either be only one IPV4 PvD for each interface or multiple IPV4 PvDs with different default gateway addresses.

7. Conveying PvD information

At the time this document was written, the conveying of PvD information was still under discussion in mif working group. Popular choices include DNS, DHCP and Route Advertisement. For PvD information provided from ISP to CE router and router to host, the approaches for PvD information delivery defined by mif may be directly used. For PvD information delivery within homenet between HNCP-enabled routers, HNCP-based approach need to be defined. The detail of how homenet could support the delivery of PvD information between routers is subjected to further discussions and will be addressed in a separate document.

8. Acknowledgements

The author would like to thank Ted Lemon, Mark Townsley, Markus Stenberg and Steven Barth for valuable discussions and contributions to this document.

9. IANA Considerations

This memo includes no request to IANA.

10. Security Considerations

TBA

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, DOI 10.17487/RFC2629, June 1999, <http://www.rfc-editor.org/info/rfc2629>.

11.2. Informative References

[RFC6418] Blanchet, M. and P. Seite, "Multiple Interfaces and Provisioning Domains Problem Statement", RFC 6418, DOI 10.17487/RFC6418, November 2011, <http://www.rfc-editor.org/info/rfc6418>.

[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>.

Authors' Addresses

Liang Geng
China Mobile

Email: gengliang@chinamobile.com

Hui Deng
China Mobile

Email: denghui@chinamobile.com

Homenet Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 6, 2015

M. Stenberg

S. Barth

March 5, 2015

Distributed Node Consensus Protocol
draft-ietf-homenet-dncp-01

Abstract

This document describes the Distributed Node Consensus Protocol (DNCP), a generic state synchronization protocol which uses Trickle and Merkle trees. DNCP is transport agnostic and leaves some of the details to be specified in profiles, which define actual implementable DNCP based protocols.

Status of This Memo

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

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 6, 2015.

Copyright Notice

Table of Contents

1. Introduction

DNCP is designed to provide a way for nodes to publish data consisting of an ordered set of TLV (Type-Length-Value) tuples and to receive the data published by all other reachable DNCP nodes.

DNCP validates the set of data within it by ensuring that it is reachable via nodes that are currently accounted for; therefore, unlike Time-To-Live (TTL) based solutions, it does not require periodic re-publishing of the data by the nodes. On the other hand, it does require the topology to be visible to every node that wants to be able to identify unreachable nodes and therefore remove old, stale data. Another notable feature is the use of Trickle to send status updates as it makes the DNCP network very thrifty when there are no updates. DNCP is most suitable for data that changes only gradually to gain the maximum benefit from using Trickle, and if more rapid state exchanges are needed, something point-to-point is recommended and just e.g. publishing of addresses of the services within DNCP.

DNCP has relatively few requirements for the underlying transport; it requires some way of transmitting either unicast datagram or stream data to a DNCP peer and, if used in multicast mode, a way of sending multicast datagrams. If security is desired and one of the built-in security methods is to be used, support for some TLS-derived transport scheme - such as TLS [RFC5246] on top of TCP or DTLS [RFC6347] on top of UDP - is also required.

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 RFC 2119 [RFC2119].

3. Terminology

A DNCP profile is a definition of a set of rules and values listed in Section 11 specifying the behavior of a DNCP based protocol, such as the used transport method. For readability, any DNCP profile specific parameters with a profile-specific fixed value are prefixed with DNCP_.

A DNCP node is a single node which runs a protocol based on a DNCP profile.

The DNCP network is a set of DNCP nodes running the same DNCP profile that can reach each other, either via discovered connectivity in the underlying network, or using each other's addresses learned via other means. As DNCP exchanges are bidirectional, DNCP nodes connected via only unidirectional links are not considered connected.

The node identifier is an opaque fixed-length identifier consisting of DNCP_NODE_IDENTIFIER_LENGTH bytes which uniquely identifies a DNCP node within a DNCP network.

A link indicates a link-layer media over which directly connected nodes can communicate.

An interface indicates a port of a node that is connected to a particular link.

A connection denotes a locally configured use of DNCP on a DNCP node, that is attached either to an interface, to a specific remote unicast address to be contacted, or to a range of remote unicast addresses that are allowed to contact.

The connection identifier is a 32-bit opaque value, which identifies a particular connection of that particular DNCP node. The value 0 is reserved for DNCP and sub-protocol purposes in the TLVs, and MUST NOT be used to identify an actual connection. This definition is in sync with [RFC3493], as the non-zero small positive integers should comfortably fit within 32 bits.

A (DNCP) peer refers to another DNCP node with which a DNCP node communicates directly on a particular connection.

The node data is a set of TLVs published by a node in the DNCP network. The whole node data is owned by the node that publishes it, and it MUST be passed along as-is, including TLVs unknown to the forwarder.

The node state is a set of metadata attributes for node data. It includes a sequence number for versioning, a hash value for comparing and a timestamp indicating the time passed since its last publication. The hash function and the number of bits used are defined in the DNCP profile.

The network state (hash) is a hash value which represents the current state of the network. The hash function and the number of bits used are defined in the DNCP profile. Whenever any node is added, removed or changes its published node data this hash value changes as well. It is calculated over the hash values of each reachable nodes' node data in ascending order of the respective node identifier.

The effective (trust) verdict for a certificate is defined as the verdict with the highest priority within the set of verdicts announced for the certificate in the DNCP network.

The neighbor graph is the undirected graph of DNCP nodes produced by retaining only bidirectional peer relationships between nodes.

4. Data Model

A DNCP node has:

- A timestamp indicating the most recent neighbor graph traversal described in Section 5.4.

A DNCP node has for every DNCP node in the DNCP network:

- A node identifier, which uniquely identifies the node.
- The node data, an ordered set of TLV tuples published by that particular node. This set of TLVs has a well-defined order based on ascending binary content (including TLV type and length). This facilitates linear time state delta processing.
- The latest update sequence number, a 32 bit number that is incremented any time the TLV set is published. For comparison purposes, a looping comparison should be used to avoid problems in case of overflow. An example would be: a < b <=> (a - b) % 2^32 & 2^31 != 0.
- The relative time (in milliseconds) since the current TLV data set with the current update sequence number was published. It is also a 32 bit number on the wire. If this number is close to overflow (greater than 2^32-2^16), a node MUST re-publish its TLVs even if there is no change to avoid overflow of the value. In other

words, absent any other changes, the TLV set MUST be re-published roughly every 49 days.

- A timestamp identifying the time it was last reachable based on neighbor graph traversal described in Section 5.4.

Additionally, a DNCP node has a set of connections for which DNCP is configured to be used. For each such connection, a node has:

- A connection identifier.
- An interface, a unicast address of a DNCP peer it should connect with, or a range of addresses from which DNCP peers are allowed to connect.
- A Trickle [RFC6206] instance with parameters I, T, and c.

For each DNCP peer detected on a connection, a DNCP node has:

- The node identifier of the DNCP peer.
- The connection identifier of the DNCP peer.
- The most recent address used by the DNCP peer (in an authenticated message, if security is enabled).

5. Operation

The DNCP protocol consists of Trickle [RFC6206] driven unicast or multicast status messages which indicate the current status of shared TLV data and additional unicast message exchanges which ensure DNCP peer reachability and synchronize the data when necessary.

If DNCP is to be used on a multicast-capable interface, as opposed to only point-to-point using unicast, a datagram-based transport which supports multicast SHOULD be defined in the DNCP profile to be used for the messages to be sent to the whole link. As this is used only to identify potential new DNCP nodes and to notify that an unicast exchange should be triggered, the multicast transport does not have to be particularly secure.

5.1. Trickle-Driven Status Update Messages

Each node MUST send either a Long Network State Update message (Section 8.2) or a Short Network State Update message (Section 8.1) every time the connection-specific Trickle algorithm [RFC6206] instance indicates that an update should be sent. The destination address of the message should be multicast in case of an interface

which is multicast-capable, or the unicast address of the remote party in case of a point-to-point connection. By default, Long Network State Update messages SHOULD be used, but if it is defined as undesirable for some case by the DNCP profile, Short Network State Update message MUST be sent instead. This may be useful to avoid fragmenting packets to multicast destinations, or for security reasons.

A Trickle state MUST be maintained separately for each connection. The Trickle state for all connections is considered inconsistent and reset if and only if the locally calculated network state hash changes. This occurs either due to a change in the local node's own node data, or due to receipt of more recent data from another node.

The Trickle algorithm has 3 parameters: Imin, Imax and k. Imin and Imax represent the minimum and maximum values for I, which is the time interval during which at least k Trickle updates must be seen on a connection to prevent local state transmission. The actual suggested Trickle algorithm parameters are DNCP profile specific, as described in Section 11.

5.2. Processing of Received Messages

This section describes how received messages are processed. The DNCP profile may specify criteria based on which received messages are ignored. Any 'reply' mentioned in the steps below denotes sending of the specified message via unicast to the originator of the message being processed. If the reply was caused by a multicast message and sent to a link with shared bandwidth it SHOULD be delayed by a random timespan in [0, Imin/2]. Sending of replies SHOULD be rate-limited by the implementation, and in case of excess load (or some other reason), a reply MAY be omitted altogether.

Upon receipt of:

Short Network State Update (Section 8.1): If the network state hash within the message differs from the locally calculated network state hash, the receiver MUST reply with a Network State Request message (Section 8.3).

Long Network State Update (Section 8.2):

* If the network state hash within the message matches the locally calculated network state hash, stop processing.

* Otherwise the receiver MUST identify all nodes for which local information is outdated (local update sequence number is lower than that within the message), potentially incorrect (local

update sequence number matches but the hash of the node data TLV differs) or missing.

* If any such nodes are identified, the receiver MUST reply with one or more Node Data Request message(s) (Section 8.4) containing Request Node Data TLV(s) (Section 9.1.2) for the corresponding nodes.

Network State Request (Section 8.3): the receiver MUST reply with a Long Network State Update (Section 8.2).

Node Data Request (Section 8.4): the receiver MUST reply with the requested data in a Node Data Reply message (Section 8.5). Optionally - if specified by the DNCP profile - multiple replies MAY be sent in order to e.g. keep size of each datagram within the PMTU to the destination. However these replies must be valid stand-alone Node Data Reply messages, with the full state for the particular nodes.

Node Data Reply (Section 8.5): If the message contains Node State TLVs that are more recent than the local state (the received TLV has a higher update sequence number, the node data TLV hash differs from the local one, or local data is missing altogether) and if the message also contains corresponding Node Data TLVs, the receiver MUST update its locally stored state.

If a message containing Node State TLVs (Section 9.2.3) is received with the node identifier matching the local node identifier and a higher update sequence number than its current local value, or the same update sequence number and a different hash, the node SHOULD re-publish its own node data with an update sequence number 1000 higher than the received one. This may occur normally once due to the local node restarting and not storing the most recently used update sequence number. If this occurs more than once, the DNCP profile should provide guidance on how to handle these situations as it indicates the existence of another active node with the same node identifier.

5.3. Adding and Removing Peers

When receiving a message on a connection from an unknown peer:

If it is a unicast message, and the message contains a Node Connection TLV (Section 9.2.1), the remote node MUST be added as a peer on the connection and a Neighbor TLV (Section 9.2.5) MUST be created for it.

If it is a multicast message, and the message contains a Node Connection TLV (Section 9.2.1), the remote node SHOULD be sent a (possibly rate-limited) unicast Network State Request Message (Section 8.3).

If keep-alives are NOT sent by the peer (either the DNCP profile does not specify the use of keep-alives or the particular peer chooses not to send keep-alive messages), some other means MUST be employed to ensure a DNCP peer is present. When the peer is no longer present, the Neighbor TLV and the local DNCP peer state MUST be removed.

5.4. Purging Unreachable Nodes

When a Neighbor TLV or a whole node is added or removed, the neighbor graph SHOULD be traversed, starting from the local node. The edges to be traversed are identified by looking for Neighbor TLVs on both nodes, that have the other node’s identifier in the neighbor node identifier, and local and neighbor connection identifiers swapped. Each node reached should be marked currently reachable.

DNCP nodes MUST be either purged immediately when not marked reachable in a particular graph traversal, or eventually after they have not been marked reachable within DNCP_GRACE_INTERVAL. During the grace period, the nodes that were not marked reachable in the most recent graph traversal MUST NOT be used for calculation of the network state hash, be provided to any applications that need to use the whole TLV graph, or be provided to remote nodes.

6. Keep-Alive Extension

The Trickle-driven messages provide a mechanism for handling of new peer detection (if applicable) on a connection, as well as state change notifications. Another mechanism may be needed to get rid of old, no longer valid DNCP peers if the transport or lower layers do not provide one.

If keep-alives are not specified in the DNCP profile, the rest of this section MUST be ignored.

A DNCP profile MAY specify either per-connection or per-peer keep-alive support.

For every connection that a keep-alive is specified for in the DNCP profile, the connection-specific keep-alive interval MUST be maintained. By default, it is DNCP_KEEPALIVE_INTERVAL. If there is a local value that is preferred for that for any reason (configuration, energy conservation, media type, ..), it should be substituted instead. If a non-default keep-alive interval is used on

any connection, a DNCP node MUST publish appropriate Keep-Alive Interval TLV(s) (Section 9.2.6).

6.1. Data Model Additions

The following additions to the Data Model (Section 4) are needed to support keep-alive:

Each node MUST have a timestamp which indicates the last time a Network State TLV (Section 9.2.2) was sent for each connection, i.e. on an interface or to the point-to-point peer(s).

Each node MUST have for each peer:

- Last consistent state timestamp: a timestamp which indicates the last time a consistent Network State TLV (Section 9.2.2) was received from the peer. When adding a new peer, it should be initialized to the current time.

6.2. Per-Connection Periodic Keep-Alive Messages

If per-connection keep-alives are enabled on connection with a multicast-enabled link, and if no traffic containing a Network State TLV (Section 9.2.2) has been sent to a particular connection within the connection-specific keep-alive interval, a Long Network State Update message (Section 8.2) or a Short Network State Update message (Section 8.1) MUST be sent on that connection. The type of message should be chosen based on the considerations in Section 5.1. The actual sending time SHOULD be further delayed by a random timespan in [0, Imin/2]. When such a message is sent, a new Trickle transmission time 't' in [I/2, I] MUST be randomly chosen.

6.3. Per-Peer Periodic Keep-Alive Messages

If per-peer keep-alives are enabled on a unicast-only connection, and if no traffic containing a Network State TLV (Section 9.2.2) has been sent to a particular peer within the connection-specific keep-alive interval, a Long Network State Update message (Section 8.2) or a Short Network State Update message (Section 8.1) MUST be sent to the peer. The type of message should be chosen based on the considerations in Section 5.1. When such a message is sent, a new Trickle transmission time 't' in [I/2, I] MUST be randomly chosen.

6.4. Received Message Processing Additions

If a message is received via unicast from the peer, the Last consistent state timestamp for the peer MUST be updated.

If the received multicast message contains a Network State TLV (Section 9.2.2) which is consistent with the locally calculated network state hash, the Last consistent state timestamp for the peer MUST be updated. If the TLV is inconsistent, and would not cause any unicast exchange, Network State Request (Section 8.3) SHOULD be sent via unicast.

6.5. Neighbor Removal

For every peer on every connection, the connection-specific keep-alive interval must be calculated by looking for Keep-Alive Interval TLVs (Section 9.2.6) published by the node, and if none exist, using the default value of DNCP_KEEPALIVE_INTERVAL. If the peer's last consistent state timestamp has not been updated for at least DNCP_KEEPALIVE_MULTIPLIER times the peer's connection-specific keep-alive interval, the Neighbor TLV for that peer and the local DNCP peer state MUST be removed.

7. Support For Dense Broadcast Links

The DNCP profile or a user configuration should limit the number of neighbors that are allowed for a (particular type of) link that a connection runs on. If that limit is exceeded, nodes without the highest Node Identifier on the link SHOULD treat the connection as an unicast connection connected to the node that has the highest Node Identifier detected on the link. The nodes MUST also keep listening to multicast traffic to both detect the presence of that node, and to react to nodes with a higher Node Identifier appearing. If the highest Node Identifier present on the link changes, the connection endpoint MUST be changed. If the Node Identifier of the local node is the highest one, the node MUST keep the connection in normal multicast mode, and the node MUST allow others to peer with it over the link.

8. Protocol Messages

For point-to-point exchanges, DNCP can run across datagram-based or reliable ordered stream-based transports. If a stream-based transport is used, a 32-bit length-value in network byte order is sent before each message to indicate the number of bytes the following message consists of.

DNCP messages are encoded as a concatenated sequence of Type-Length-Value objects (Section 9). In order to facilitate fast comparing of local state with that in a received message update, all TLVs in every encoding scope (either within the message itself, or within a container TLV) MUST be placed in ascending order based on the binary comparison of both TLV header and value. By design, the TLVs which

MUST be present have the lowest available type values, ensuring they will naturally occur at the start of the Protocol Message, resembling a fixed format header.

DNCP profiles MAY add additional TLVs to the message specified here, or even define additional messages as needed. TLVs not recognized by the receiver MUST be ignored.

8.1. Short Network State Update Message

The Short Network State Update Message is used to announce the sender’s view of the network state using multicast.

The following TLVs MUST be present:

- One Node Connection TLV (Section 9.2.1) identifying the originating node and connection.
- One Network State TLV (Section 9.2.2) containing the network state hash as calculated by the sender.

The Short Network Status update message MUST NOT contain any Node State TLV(s) (Section 9.2.3).

8.2. Long Network State Update Message

The Long Network State Update Message is used to announce the sender’s view of the network state and all node states using multicast or unicast.

The following TLVs MUST be present:

- One Node Connection TLV (Section 9.2.1) identifying the originating node and connection.
- One Network State TLV (Section 9.2.2) containing the network state hash as calculated by the sender.
- One or more Node State TLVs (Section 9.2.3) containing the node state of DNCP nodes as currently known to the sender.

The Long Network State Update message MUST include the corresponding Node State TLV (Section 9.2.3) for each Node Data TLV used to calculate the network state hash.

8.3. Network State Request Message

The Network State Request message is used to request the recipient's view of the network state and all node states currently known to it.

The following TLVs MUST be present:

- One Request Network State TLV (Section 9.1.1) indicating the type of request.

8.4. Node Data Request Message

The Node Data Request message is used to request the node state and data of one or more DNCP nodes in the network.

The following TLVs MUST be present:

- One or more Request Node Data TLVs (Section 9.1.2) indicating the nodes for which state and data is requested.

8.5. Node Data Reply Message

The Node Data Request message is used to provide the node data of one or more DNCP nodes in the network.

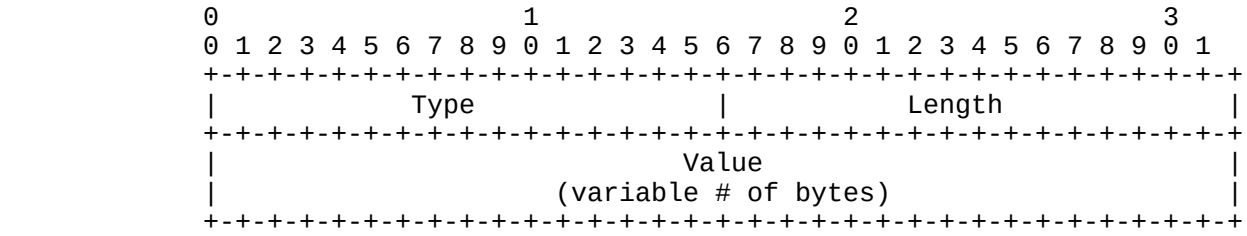
The following TLVs MUST be present:

- One Node Connection TLV (Section 9.2.1) identifying the originating node and connection.

- One or more Node State TLV (Section 9.2.3) and Node Data TLV (Section 9.2.4) pairs with matching node identifiers for each node previously requested in a Node Data Request message (Section 8.4).

9. Type-Length-Value Objects

Each TLV is encoded as a 2 byte type field, followed by a 2 byte length field (of the value, excluding header; 0 means no value) followed by the value itself (if any). Both type and length fields in the header as well as all integer fields inside the value - unless explicitly stated otherwise - are represented in network byte order. Zero padding bytes MUST be added up to the next 4 byte boundary if the length is not divisible by 4. These padding bytes MUST NOT be included in the length field.



For example, type=123 (0x7b) TLV with value 'x' (120 = 0x78) is encoded as: 007B 0001 7800 0000.

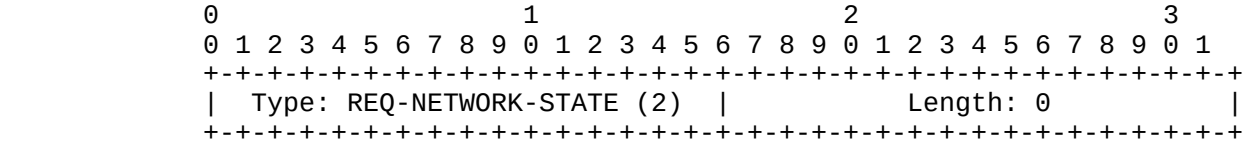
Notation:

.. = octet string concatenation operation.

H(x) = non-cryptographic hash function specified by DNCP profile.

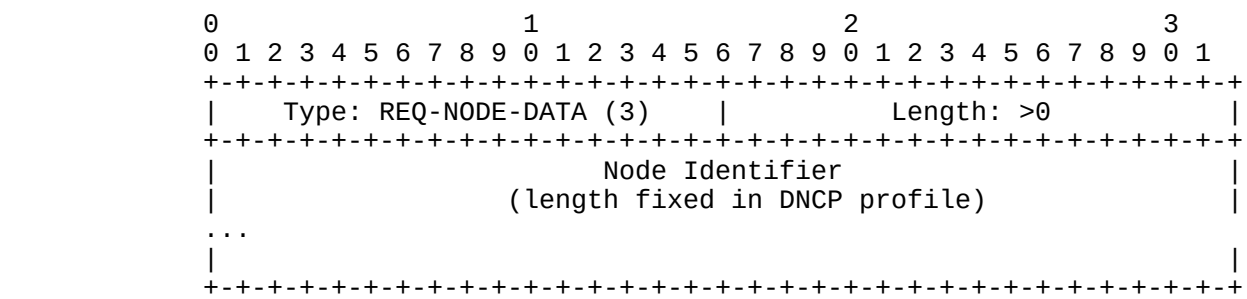
9.1. Request TLVs

9.1.1. Request Network State TLV



This TLV is used to identify a Network State Request message (Section 8.3).

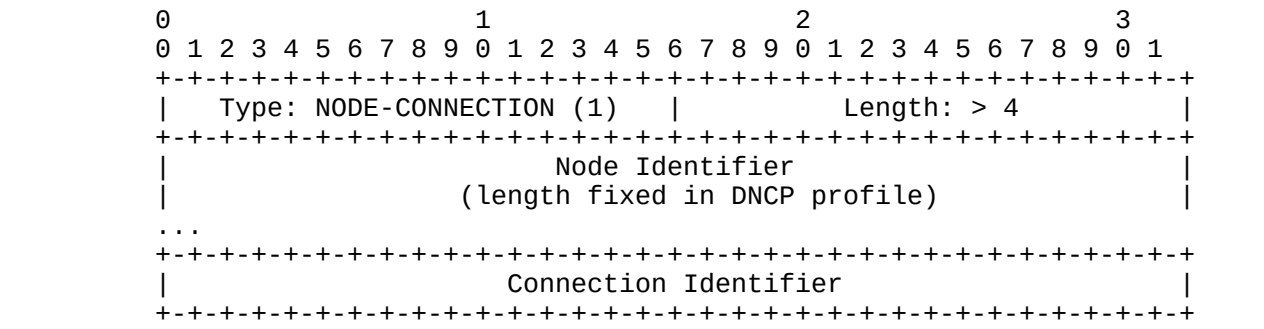
9.1.2. Request Node Data TLV



This TLV is used within a Node Data Request message (Section 8.4) to request node state and node data for the node with matching node identifier, if any, to be included in a subsequent Node Data Reply message (Section 8.5).

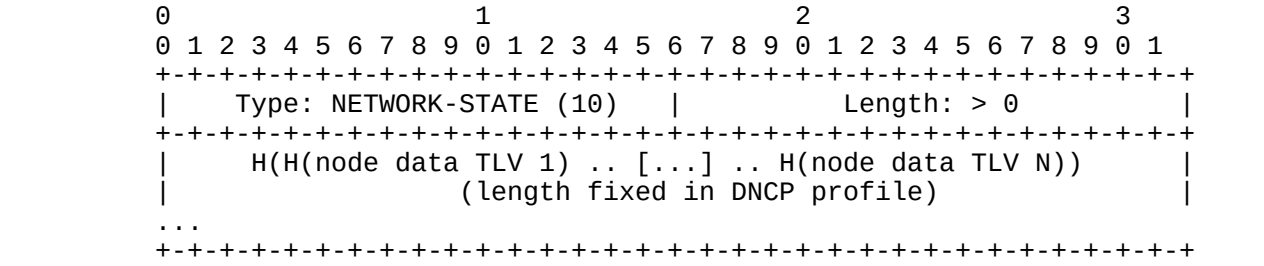
9.2. Data TLVs

9.2.1. Node Connection TLV



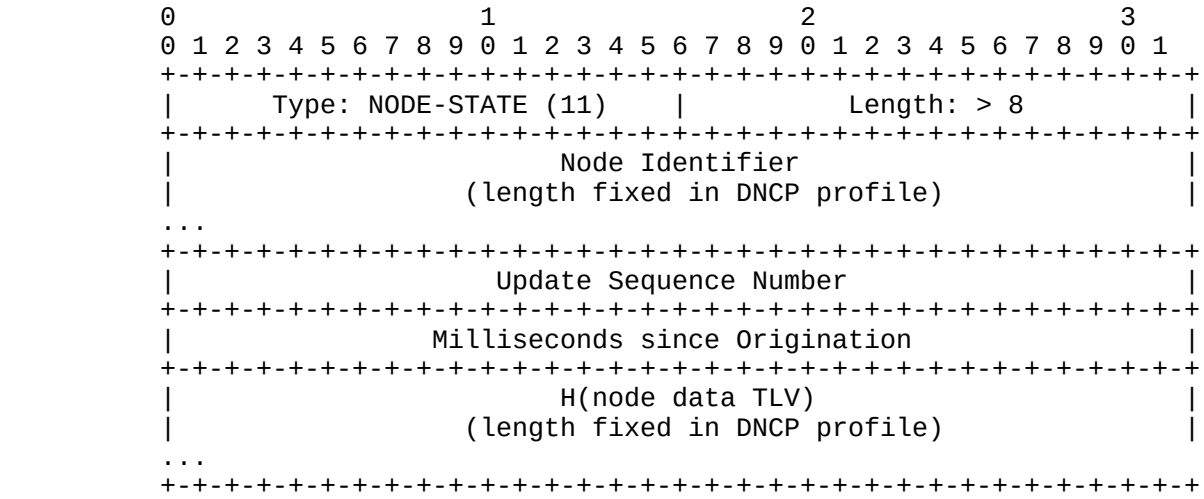
This TLV identifies both the local node's node identifier, as well as the particular connection identifier. It MUST be sent in all messages if bidirectional peer relationship is desired with remote nodes. Bidirectional peer relationship is not necessary for read-only access to the DNCP state, but it is required to be able to publish something.

9.2.2. Network State TLV



This TLV contains the current locally calculated network state hash. The network state hash is derived by calculating the hash value for each currently reachable node's Node Data TLV, concatenating said hash values based on the ascending order of their corresponding node identifiers, and hashing the resulting concatenated hash values.

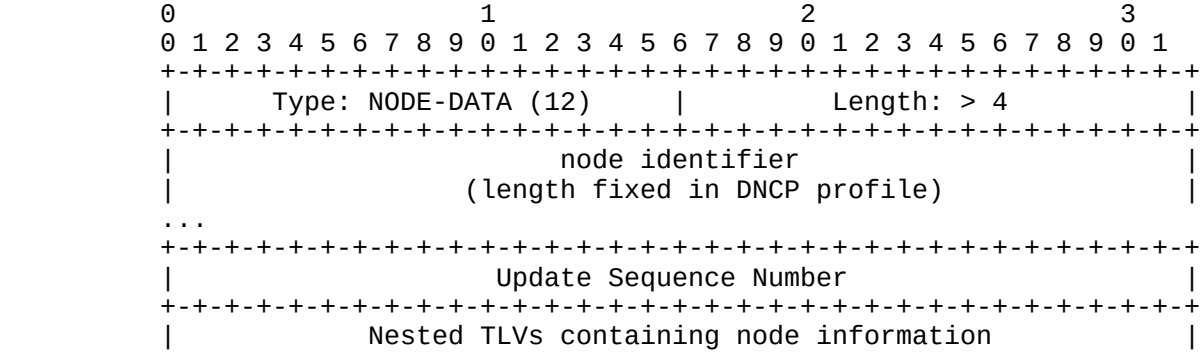
9.2.3. Node State TLV



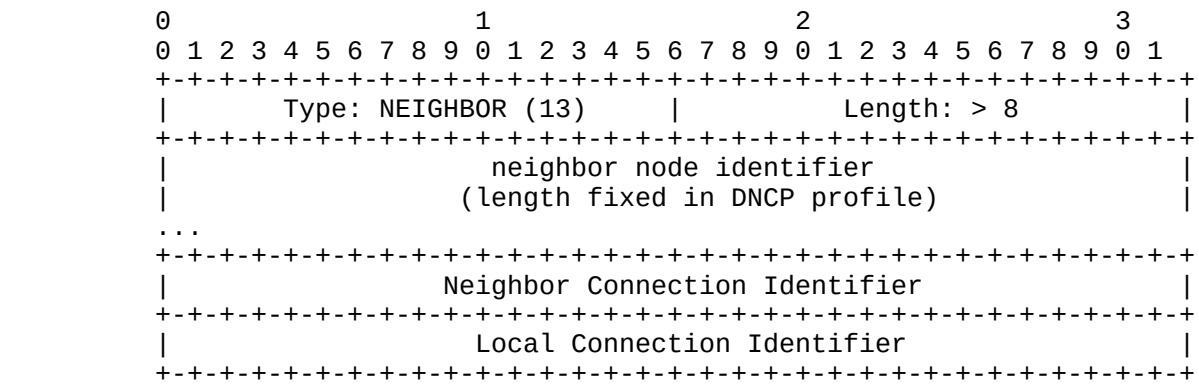
This TLV represents the local node's knowledge about the published state of a node in the DNCP network identified by the node identifier field in the TLV.

The whole network should have roughly same idea about the time since origination of any particular published state. Therefore every node, including the originating one, MUST increment the time whenever it needs to send a Node State TLV for an already published Node Data TLV. This age value is not included within the Node Data TLV, however, as that is immutable and used to detect changes in the network state.

9.2.4. Node Data TLV

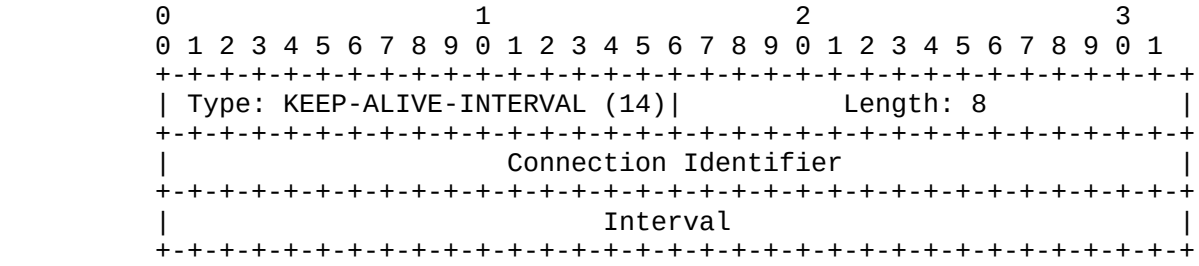


9.2.5. Neighbor TLV (within Node Data TLV)



This TLV indicates that the node in question vouches that the specified neighbor is reachable by it on the specified local connection. The presence of this TLV at least guarantees that the node publishing it has received traffic from the neighbor recently. For guaranteed up-to-date bidirectional reachability, the existence of both nodes' matching Neighbor TLVs should be checked.

9.2.6. Keep-Alive Interval TLV (within Node Data TLV)

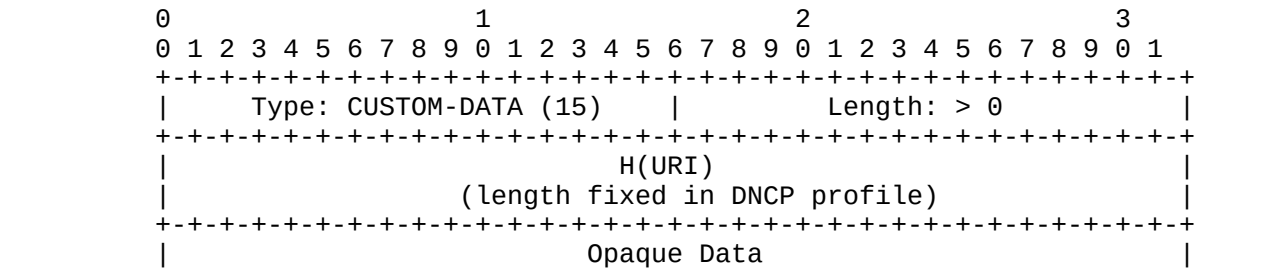


This TLV indicates a non-default interval being used to send keep-alive messages specified in Section 6.

Connection identifier is used to identify the particular connection for which the interval applies. If 0, it applies for ALL connections for which no specific TLV exists.

Interval specifies the interval in milliseconds at which the node sends keep-alives. A value of zero means no keep-alives are sent at all; in that case, some lower layer mechanism that ensures presence of nodes MUST be available and used.

9.3. Custom TLV (within/without Node Data TLV)



This TLV can be used to contain anything; the URI used should be under control of the author of that specification. For example:

V = H('http://example.com/author/json-for-dncp') .. '{"cool": "json extension!"}'

or

V = H('mailto:author@example.com') .. '{"cool": "json extension!"}'

10. Security and Trust Management

If specified in the DNCP profile, either DTLS [RFC6347] or TLS [RFC5246] may be used to authenticate and encrypt either some (if specified optional in the profile), or all unicast traffic. The following methods for establishing trust are defined, but it is up to the DNCP profile to specify which ones may, should or must be supported.

10.1. Pre-Shared Key Based Trust Method

A PSK-based trust model is a simple security management mechanism that allows an administrator to deploy devices to an existing network by configuring them with a pre-defined key, similar to the configuration of an administrator password or WPA-key. Although limited in nature it is useful to provide a user-friendly security mechanism for smaller networks.

10.2. PKI Based Trust Method

A PKI-based trust-model enables more advanced management capabilities at the cost of increased complexity and bootstrapping effort. It however allows trust to be managed in a centralized manner and is therefore useful for larger networks with a need for an authoritative trust management.

10.3. Certificate Based Trust Consensus Method

The certificate-based consensus model is designed to be a compromise between trust management effort and flexibility. It is based on X.509-certificates and allows each DNCP node to provide a verdict on any other certificate and a consensus is found to determine whether a node using this certificate or any certificate signed by it is to be trusted.

The current effective trust verdict for any certificate is defined as the one with the highest priority from all verdicts announced for said certificate at the time.

10.3.1. Trust Verdicts

Trust Verdicts are statements of DNCP nodes about the trustworthiness of X.509-certificates. There are 5 possible verdicts in order of ascending priority:

0 Neutral : no verdict exists but the DNCP network should determine one.

1 Cached Trust : the last known effective verdict was Configured or Cached Trust.

2 Cached Distrust : the last known effective verdict was Configured or Cached Distrust.

3 Configured Trust : trustworthy based upon an external ceremony or configuration.

4 Configured Distrust : not trustworthy based upon an external ceremony or configuration.

Verdicts are differentiated in 3 groups:

- Configured verdicts are used to announce explicit verdicts a node has based on any external trust bootstrap or predefined relation a node has formed with a given certificate.
- Cached verdicts are used to retain the last known trust state in case all nodes with configured verdicts about a given certificate have been disconnected or turned off.
- The Neutral verdict is used to announce a new node intending to join the network so a final verdict for it can be found.

The current effective trust verdict for any certificate is defined as the one with the highest priority within the set of verdicts + announced for the certificate in the DNCP network. A node MUST be trusted for participating in the DNCP network if and only if the current effective verdict for its own certificate or any one in its certificate hierarchy is (Cached or Configured) Trust and none of the certificates in its hierarchy have an effective verdict of (Cached or Configured) Distrust. In case a node has a configured verdict, which is different from the current effective verdict for a certificate, the current effective verdict takes precedence in deciding trustworthiness. Despite that, the node still retains and announces its configured verdict.

10.3.2. Trust Cache

Each node SHOULD maintain a trust cache containing the current effective trust verdicts for all certificates currently announced in the DNCP network. This cache is used as a backup of the last known state in case there is no node announcing a configured verdict for a known certificate. It SHOULD be saved to a non-volatile memory at reasonable time intervals to survive a reboot or power outage.

Every time a node (re)joins the network or detects the change of an effective trust verdict for any certificate, it will synchronize its cache, i.e. store new effective verdicts overwriting any previously cached verdicts. Configured verdicts are stored in the cache as their respective cached counterparts. Neutral verdicts are never stored and do not override existing cached verdicts.

10.3.3. Announcement of Verdicts

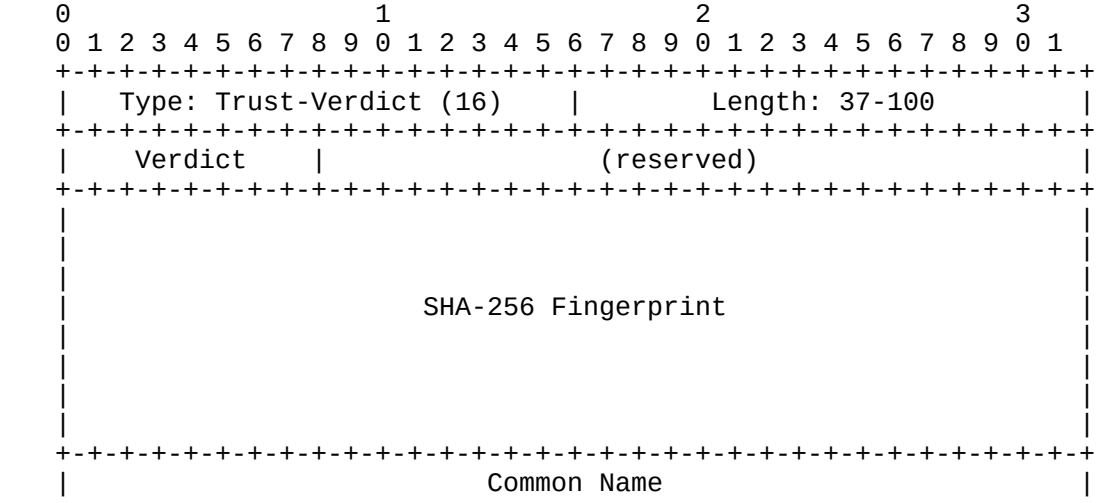
A node SHOULD always announce any configured trust verdicts it has established by itself, and it MUST do so if announcing the configured trust verdict leads to a change in the current effective verdict for the respective certificate. In absence of configured verdicts, it MUST announce cached trust verdicts it has stored in its trust cache, if one of the following conditions applies:

- The stored verdict is Cached Trust and the current effective verdict for the certificate is Neutral or does not exist.
- The stored verdict is Cached Distrust and the current effective verdict for the certificate is Cached Trust.

A node rechecks these conditions whenever it detects changes of announced trust verdicts anywhere in the network.

Upon encountering a node with a hierarchy of certificates for which there is no effective verdict, a node adds a Neutral Trust-Verdict-TLV to its node data for all certificates found in the hierarchy, and publishes it until an effective verdict different from Neutral can be found for any of the certificates, or a reasonable amount of time (10 minutes is suggested) with no reaction and no further authentication attempts has passed. Such verdicts SHOULD also be limited in rate and number to prevent denial-of-service attacks.

Trust verdicts are announced using Trust-Verdict TLVs:



Verdict represents the numerical index of the verdict.

(reserved) is reserved for future additions and MUST be set to 0 when creating TLVs and ignored when parsing them.

SHA-256 Fingerprint contains the SHA-256 [RFC6234] hash value of the certificate in DER-format.

Common Name contains the variable-length (1-64 bytes) common name of the certificate. Final byte MUST have value of 0.

10.3.4. Bootstrap Ceremonies

The following non-exhaustive list of methods describes possible ways to establish trust relationships between DNCP nodes and node certificates. Trust establishment is a two-way process in which the existing network must trust the newly added node and the newly added node must trust at least one of its neighboring nodes. It is therefore necessary that both the newly added node and an already

trusted node perform such a ceremony to successfully introduce a node into the DNCP network. In all cases an administrator MUST be provided with external means to identify the node belonging to a certificate based on its fingerprint and a meaningful common name.

10.3.4.1. Trust by Identification

A node implementing certificate-based trust MUST provide an interface to retrieve the current set of effective trust verdicts, fingerprints and names of all certificates currently known and set configured trust verdicts to be announced. Alternatively it MAY provide a companion DNCP node or application with these capabilities with which it has a pre-established trust relationship.

10.3.4.2. Preconfigured Trust

A node MAY be preconfigured to trust a certain set of node or CA certificates. However such trust relationships MUST NOT result in unwanted or unrelated trust for nodes not intended to be run inside the same network (e.g. all other devices by the same manufacturer).

10.3.4.3. Trust on Button Press

A node MAY provide a physical or virtual interface to put one or more of its internal network interfaces temporarily into a mode in which it trusts the certificate of the first DNCP node it can successfully establish a connection with.

10.3.4.4. Trust on First Use

A node which is not associated with any other DNCP node MAY trust the certificate of the first DNCP node it can successfully establish a connection with. This method MUST NOT be used when the node has already associated with any other DNCP node.

11. DNCP Profile-Specific Definitions

Each DNCP profile MUST define following:

- How the messages are secured: Not at all, optionally or always with the TLS scheme defined here using one or more of the methods, or with something else. If the links with DNCP nodes can be sufficiently secured or isolated, it is possible to run DNCP in a secure manner without using any form of authentication or encryption.

- Unicast and optionally multicast transport protocol(s) to be used. If TLS scheme within this document is to be used security, TLS or

DTLS support for at least the unicast transport protocol is mandatory.

- Transport protocols' parameters such as port numbers to be used, or multicast address to be used. Unicast, multicast, and secure unicast may each require different parameters, if applicable.

- When receiving messages, what sort of messages are dropped, as specified in Section 5.2.

- What is the criteria for sending Trickle-based Long Network State Update message (Section 8.2) on an interface or to a DNCP peer.

- How to deal with node identifier collision as described in Section 5.2. Main options are either for one or both nodes to assign new node identifiers to themselves, or to notify someone about a fatal error condition in the DNCP network.

- Imin, Imax and k ranges to be suggested for implementations to be used in the Trickle algorithm. The Trickle algorithm does not require these to be same across all implementations for it to work, but similar orders of magnitude helps implementations of a DNCP profile to behave more consistently and to facilitate estimation of lower and upper bounds for behavior of the network.

- Hash function H(x) to be used, and how many bits of the input are actually used. The chosen hash function is used to handle both hashing of node specific data, and network state hash, which is a hash of node specific data hashes. SHA-256 defined in [RFC6234] is the recommended default choice.

- DNCP_NODE_IDENTIFIER_LENGTH: The fixed length of a node identifier (in bytes).

- DNCP_GRACE_INTERVAL: How long node data for unreachable nodes is kept.

- Whether to send keep-alives, and if so, on an interface, using multicast, or directly using unicast to peers. Keep-alive has also associated parameters:

  * DNCP_KEEPALIVE_INTERVAL: How often keep-alive messages are to be sent by default (if enabled).

  * DNCP_KEEPALIVE_MULTIPLIER: How many times the DNCP_KEEPALIVE_INTERVAL (or peer-supplied keep-alive interval value) a node may not be heard from to be considered still valid.

12. Security Considerations

DNCP profiles may use multicast to indicate DNCP state changes and for keep-alive purposes. However, no actual data TLVs will be sent across that channel. Therefore an attacker may only learn hash values of the state within DNCP and may be able to trigger unicast synchronization attempts between nodes on a local link this way. A DNCP node should therefore rate-limit its reactions to multicast packets.

When using DNCP to bootstrap a network, PKI based solutions may have issues when validating certificates due to potentially unavailable accurate time, or due to inability to use the network to either check Certifcate Revocation Lists or perform on-line validation.

The Certificate-based trust consensus mechanism defined in this document allows for a consenting revocation, however in case of a compromised device the trust cache may be poisoned before the actual revocation happens allowing the distrusted device to rejoin the network using a different identity. Stopping such an attack might require physical intervention and flushing of the trust caches.

13. IANA Considerations

IANA should set up a registry for DNCP TLV types, with the following initial contents:

0: Reserved (should not happen on wire)

1: Node connection

2: Request network state

3: Request node data

4-9: Reserved for DNCP profile use

10: Network state

11: Node state

12: Node data

13: Neighbor

14: Keep-alive interval

15: Custom

16: Trust-Verdict

17-31: Reserved for future DNCP versions.

192-255: Reserved for per-implementation experimentation. The nodes using TLV types in this range SHOULD use e.g. Custom TLV to identify each other and therefore eliminate potential conflict caused by potential different use of same TLV numbers.

For the rest of the values (32-191, 256-65535), policy of 'standards action' should be used.

14. References

14.1. Normative references

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

[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.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

14.2. Informative references

[RFC3493] Gilligan, R., Thomson, S., Bound, J., McCann, J., and W. Stevens, "Basic Socket Interface Extensions for IPv6", RFC 3493, February 2003.

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

Appendix A. Some Outstanding Issues

Should better forms of combined messages be defined? e.g. allow sending both request-network-state, and currently set of known local state at same time in one message.

Should some sort of fragmentation scheme be defined for the data? Currently, DNCP uses Merkle tree of depth 2 (node data -> node hash -> network hash). However, it essentially treats all TLVs published by a single node as a single chunk on the protocol level. Is that a

scalability problem? Adding one more level to the tree might address this.

Appendix B. Some Obvious Questions and Answers

Q: Should there be nested container syntax that is actually self-describing? (i.e. type flag that indicates container, no body except sub-TLVs?)

A: Not for now, but perhaps valid design.. TBD.

Q: Add third case for multicast - 'medium' network state, which is 'long' one, but partial?

A: Drops typical convergence on large networks 5->3 packets, at expense of some specification/implementation complexity. However, as anything else than short network state leaks information via multicast, it does not seem worth it as secure protocols probably want to prevent multicast sending of anything else than short network state in any case. Additionally, the long network state being complete set of nodes actually facilitates light-weight nodes that do not want to do graph-based pruning. So all in all, perhaps not worth it.

Q: 32-bit connection id?

A: Here, it would save 32 bits per neighbor if it was 16 bits (and less is not realistic). However, TLVs defined elsewhere would not seem to even gain that much on average. 32 bits is also used for ifindex in various operating systems, making for simpler implementation.

Q: Why have topology information at all?

A: It is an alternative to the more traditional seq#/TTL-based flooding schemes. In steady state, there is no need to e.g. re-publish every now and then.

Appendix C. Changelog

draft-ietf-homenet-dncp-01:

- Fixed keep-alive semantics to consider unicast requests also updates of most recently consistent, and added proactive unicast request to ensure even inconsistent keep-alive messages eventually triggering consistency timestamp update.

- Facilitated (simple) read-only clients by making Node Connection TLV optional if just using DNCP for read-only purposes.

- Added text describing how to deal with "dense" networks, but left actual numbers and mechanics up to DNCP profiles and (local) configurations.

draft-ietf-homenet-dncp-00: Split from pre-version of draft-ietf-homenet-hncp-03 generic parts. Changes that affect implementations:

- TLVs were renumbered.

- TLV length does not include header (=-4). This facilitates e.g. use of DHCPv6 option parsing libraries (same encoding), and reduces complexity (no need to handle error values of length less than 4).

- Trickle is reset only when locally calculated network state hash is changes, not as remote different network state hash is seen. This prevents e.g. attacks by multicast with one multicast packet to force Trickle reset on every interface of every node on a link.

- Instead of 'ping', use 'keep-alive' (optional) for dead peer detection. Different message used!

Appendix D. Draft Source

As usual, this draft is available at https://github.com/fingon/ietf-drafts/ in source format (with nice Makefile too). Feel free to send comments and/or pull requests if and when you have changes to it!

Appendix E. Acknowledgements

Thanks to Ole Troan, Pierre Pfister, Mark Baugher, Mark Townsley, Juliusz Chroboczek and Jiazi Yi for their contributions to the draft.

Authors' Addresses

Markus Stenberg
Helsinki 00930
Finland

Email: markus.stenberg@iki.fi

Steven Barth
Halle 06114
Germany

Email: cyrus@openwrt.org

HOMENET
Internet-Draft
Intended status: Standards Track
Expires: August 20, 2015

D. Migault (Ed)
Ericsson
W. Cloetens
SoftAtHome
C. Griffiths
Dyn
R. Weber
Nominum

February 16, 2015

Outsourcing Home Network Authoritative Naming Service
draft-ietf-homenet-front-end-naming-delegation-01.txt

Abstract

CPEs are designed to provide IP connectivity to home networks. Most CPEs assign IP addresses to the nodes of the home network which makes it a good candidate for hosting the naming service. With IPv6, the naming service makes nodes reachable from the home network as well as from the Internet.

However, CPEs have not been designed to host such a naming service exposed on the Internet. This may expose the CPEs to resource exhaustion which would make the home network unreachable, and most probably would also affect the home network inner communications.

In addition, DNSSEC management and configuration may not be well understood or mastered by regular end users. Misconfiguration may also results in naming service disruption, thus these end users may prefer to rely on third party naming providers.

This document describes a homenet naming architecture where the CPEs manage the DNS zone associates to its home network, and outsources the naming service and eventually the DNSSEC management on the Internet to a third party designated as the Public Authoritative Servers.

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/.

This Internet-Draft will expire on August 20, 2015.

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. To access services hosted in the home network with IPv6 addresses, end users prefer to use names instead of long and complex IPv6 addresses.

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 the CPEs a good candidate to manage binding between names and IP addresses of the nodes. In addition, [RFC7368] recommends that home networks be resilient to connectivity disruption from the ISP. This requires that a dedicate device inside the home network manage bindings between names and IP addresses of the nodes and builds the DNS Homenet Zone. All this makes the CPE the natural candidate for setting the DNS(SEC) zone file of the home network.

CPEs are usually low powered devices designed for the home network, but not for heavy traffic. As a result, hosting the an authoritative DNS service on the Internet may expose the home network to resource exhaustion, which may isolate the home network from the Internet and affect the services hosted by the CPEs, thus affecting the overall home network communications.

In order to avoid resource exhaustion, this document describes an architecture that outsources the authoritative naming service of the home network. More specifically, the DNS(SEC) Homenet Zone built by the CPE is outsourced to Public Authoritative Servers. These servers publish the corresponding DN(SEC) Public Zone on the Internet. Section 4.1 describes the architecture. In order to keep the DNS(SEC) Public Zone up-to-date Section 5 describes how the DNS(SEC) Homenet Zone and the DN(SEC) Public Zone can be synchronized. The proposed architecture aims at deploying DNSSEC and the DNS(SEC) Public Zone is expected to be signed with a secure delegation. The zone signing and secure delegation can be performed either by the CPE or by the Public Authoritative Servers. Section 6 discusses these

two alternatives. Section 7 discusses multiple views aspects and provide guidance to avoid them. Section 8 discusses the case of the reverse zone. Section 9 and Section 10 respectively discuss privacy and security considerations when outsourcing the DNS Homenet Zone.

3. Terminology

- Customer Premises Equipment: (CPE) is the router providing connectivity to the home network. It is configured and managed by the end user. In this document, the CPE MAY also hosts services such as DHCPv6. This device MAY be provided by the ISP.

- Registered Homenet Domain: is the Domain Name associated to the home network.

- DNS Homenet Zone: is the DNS zone associated to the home network. This zone is set by the CPE and essentially contains the bindings between names and IP addresses of the nodes of the home network. In this document, the CPE does neither perform any DNSSEC management operations such as zone signing nor provide an authoritative service for the zone. Both are delegated to the Public Authoritative Server. The CPE synchronizes the DNS Homenet Zone with the Public Authoritative Server via a hidden master / slave architecture. The Public Authoritative Server MAY use specific servers for the synchronization of the DNS Homenet Zone: the Public Authoritative Name Server Set as public available name servers for the Registered Homenet Domain.

- DNS Homenet Reverse Zone: The reverse zone file associated to the DNS Homenet Zone.

- Public Authoritative Server: performs DNSSEC management operations as well as provides the authoritative service for the zone. In this document, the Public Authoritative Server synchronizes the DNS Homenet Zone with the CPE via a hidden master / slave architecture. The Public Authoritative Server acts as a slave and MAY use specific servers called Public Authoritative Name Server Set. Once the Public Authoritative Server synchronizes the DNS Homenet Zone, it signs the zone and generates the DNSSEC Public Zone. Then the Public Authoritative Server hosts the zone as an authoritative server on the Public Authoritative Master(s).

- DNSSEC Public Zone: corresponds to the signed version of the DNS Homenet Zone. It is hosted by the Public Authoritative Server,

which is authoritative for this zone, and is reachable on the Public Authoritative Master(s).

- Public Authoritative Master(s): are the visible name server hosting the DNSSEC Public Zone. End users' resolutions for the Homenet Domain are sent to this server, and this server is a master for the zone.

- Public Authoritative Name Server Set: is the server the CPE synchronizes the DNS Homenet Zone. It is configured as a slave and the CPE acts as master. The CPE sends information so the DNSSEC zone can be set and served.

- Reverse Public Authoritative Master(s): are the visible name server hosting the DNS Homenet Reverse Zone. End users' resolutions for the Homenet Domain are sent to this server, and this server is a master for the zone.

- Reverse Public Authoritative Name Server Set: is the server the CPE synchronizes the DNS Homenet Reverse Zone. It is configured as a slave and the CPE acts as master. The CPE sends information so the DNSSEC zone can be set and served.

4. Architecture Description

This section describes the architecture for outsourcing the authoritative naming service from the CPE to the Public Authoritative Master(s). Section 4.1 describes the architecture, Section 4.2 and Section 4.3 illustrate this architecture and shows how the DNS(SEC) Homenet Zone should be built by the CPE, as well as lists the necessary parameters the CPE needs to outsource the authoritative naming service. These two section 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 DNS(SEC) Homenet Zone associated to the home network. How the DNS(SEC) Homenet Zone is built is out of the scope of this document. The CPE may host and involve multiple services like a web GUI, DHCP [RFC6644] or mDNS [RFC6762]. These services may coexist and may be used to populate the DNS Homenet Zone. This document assumes the DNS(SEC) Homenet Zone has been populated with domain names that are intended to be publicly published and that are publicly reachable. More specifically, names associated to 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 DNS(SEC) Homenet Zone.

Once the DNS(SEC) Homenet Zone has been built, the CPE does not host the authoritative naming service for it, but instead outsources it to the Public Authoritative Servers. The Public Authoritative Servers take the DNS(SEC) Homenet as an input and publishes the DNS(SEC) Public Zone. In fact the DNS(SEC) Homenet Zone and the DNS(SEC) Public Zone have different names as they may be different. If the CPE does not sign the DNS Homenet Zone, for example, the Public Authoritative Servers may instead sign it on behalf of the CPE. Figure 1 provides a more detailed description of the Public Authoritative Servers, but overall, it is expected that the CPE provides the DNS(SEC) Homenet Zone, the DNS(SEC) Public Zone is derived from the DNS(SEC) Homenet Zone and published on the Internet.

As a result, DNS(SEC) queries from the DNS(SEC) Resolvers on the Internet are answered by the Public Authoritative Server and do not reach the CPE. Figure 1 illustrates the case of the resolution of node1.example.com.

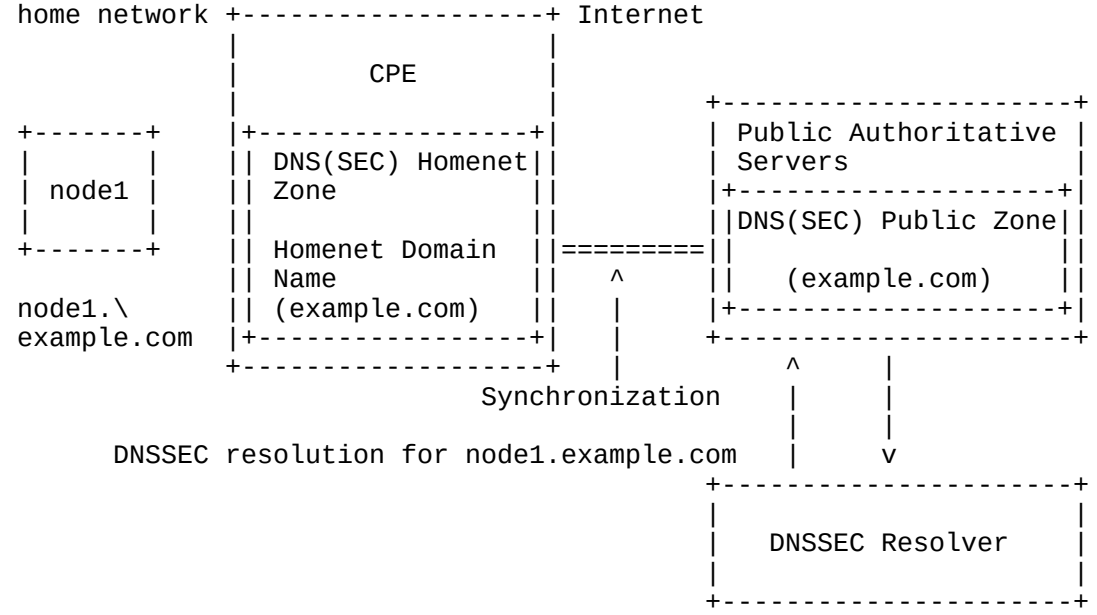


Figure 1: Homenet Naming Architecture Description

The Public Authoritative Servers are described in Figure 2. The Public Authoritative Name Server Set receives the DNS(SEC) Homenet

Zone as an input. The received zone may be transformed to output the DNS(SEC) Public Zone. Various operations may be performed here, however this document only considers zone signing as potential operation. This could occur only when the CPE outsources this operation to the Public Authoritative Name Server Set. On the other hand, if the CPE signs the DNSSEC Homenet Zone itself, the zone it collected by the Public Authoritative Name Server Set and directly transferred to the Public Authoritative Master. Implications of such policy are detailed in Section 6 and Section 7.

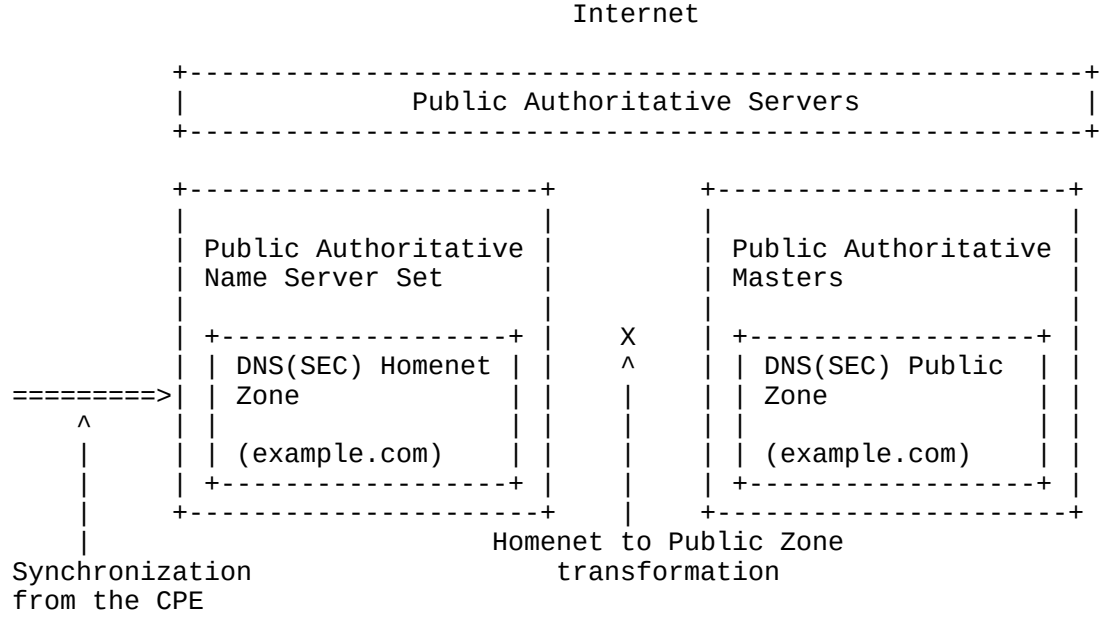


Figure 2: Public Authoritative Servers Description

4.2. Example: DNS(SEC) Homenet Zone

This section is not normative and intends to illustrate how the CPE builds the DNS(SEC) Homenet Zone.

As depicted in Figure 1 and Figure 2, the DNS(SEC) Public Zone is hosted on the Public Authoritative Masters, whereas the DNS(SEC) 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 Masters. In other words "Homenet to Public Zone transformation" is the identity.

In that case, the DNS Homenet Zone should configure its Name Server RRset (NS) and Start of Authority (SOA) with the ones associated to the Public Authoritative Masters. This is illustrated in Figure 3. public.masters.example.net is the FQDN of the Public Authoritative Masters, and IP1, IP2, IP3, IP4 are the associated IP addresses. Then the CPE should add the different new nodes that enter the home network, remove those that should be removed and sign the DNS Homenet Zone.

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

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

@   NS  public.authoritative.servers.example.net

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

Figure 3: DNS 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 Master and the Public Authoritative Name Server Set and published on the DNS Public Authoritative Master.

- MNAME: indicates the primary master. In our case the zone is published on the Public Authoritative Master, and its name MUST be mentioned. If multiple Public Authoritative Masters are involved, one of them MUST be chosen. More specifically, the CPE MUST NOT place the name of the Hidden Master.

- RNAME: indicates the email address to reach the administrator. [RFC2142] recommends to use hostmaster@domain and replacing the '@' sign by '.'.

- REFRESH and RETRY: indicate respectively in seconds how often slaves need to check the master and the time between two

refresh when a refresh has failed. Default value indicated by [RFC1033] are 3600 (1 hour) for refresh and 600 (10 minutes) for retry. This value MAY be long for highly dynamic content. However, Public Authoritative Masters and the CPE are expected to implement NOTIFY [RFC1996]. Then short values MAY increase the bandwidth usage for slaves hosting large number of zones. As a result, default values looks fine.

EXPIRE: is the upper limit data SHOULD be kept in absence of refresh. Default value indicated by [RFC1033] is 3600000 about 42 days. In home network architectures, the CPE provides both the DNS synchronization and the access to the home network. This device MAY be plug / unplugged by the end user without notification, thus we recommend large period.

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

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.

Public Authoritative Name Server Set may be defined with the following parameters. These parameters are necessary to establish a secure channel between the CPE and the Public Authoritative Name Server Set:

- Public Authoritative Name Server Set: The associated FQDNs or IP addresses of the Public Authoritative 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 Public Server. This MAY depend on the implementation but we should consider 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, files for the CPE private keys, certificates and certification authority should be specified.

- Public Authoritative Master(s): The FQDN or IP addresses of the Public Authoritative Master. It MAY correspond to the data

that will be set in the NS RRsets and SOA of the DNS Homenet Zone. 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.

- Registered Homenet Domain: The domain name the Public Authoritative is configured for DNS slave, DNSSEC zone signing and DNSSEC zone hosting.

Setting the DNS(SEC) Homenet Zone requires the following information.

- Registered Homenet Domain: The Domain Name of the zone. Multiple Registered Homenet Domain may be provided. This will generate the creation of multiple DNS Homenet Zones.

- Public Authoritative Server: The Public Authoritative Servers associated to the Registered Homenet Domain. Multiple Public Authoritative Server may be provided.

5. Synchronization between CPE and Public Authoritative Servers

The DNS(SEC) Homenet Reverse Zone and the DNS Homenet Zone can be updated either with DNS update [RFC2136] or using a master / slave synchronization. The master / slave mechanism is preferred as it better scales and avoids DoS attacks: First the master notifies the slave the zone must be updated, and leaves the slave to proceed to the update when possible. Then, the NOTIFY message sent by the master is a small packet that is less likely to load the slave. At last, the AXFR query performed by the slave is a small packet sent over TCP (section 4.2 [RFC5936]) which makes unlikely the slave to perform reflection attacks with a forged NOTIFY. On the other hand, DNS updates can use UDP, packets require more processing then a NOTIFY, and they do not provide the server the opportunity to post-pone the update.

This document recommends the use of a master / slave mechanism instead of the use of nsupdates. This section details the master / slave mechanism.

5.1. Synchronization with a Hidden Master

Uploading and dynamically updating the zone file on the Public Authoritative Name Server Set can be seen as zone provisioning between the CPE (Hidden Master) and the Public Authoritative Name Server Set (Slave Server). This can be handled either in band or out of band.

The Public Authoritative Name Server Set is configured as a slave for the Homenet Domain Name. This slave configuration has been previously agreed between the end user and the provider of the Public Authoritative Servers. In order to set the master/ slave architecture, the CPE acts as a Hidden Master Server, which is a regular Authoritative DNS(SEC) Server listening on the WAN interface.

The Hidden Master Server is expected to accept SOA [RFC1033], AXFR [RFC1034], and IXFR [RFC1995] queries from its configured slave DNS servers. The Hidden Master Server SHOULD send NOTIFY messages [RFC1996] in order to update Public DNS server zones as updates occur. Because, DNS Homenet Zones are likely to be small, CPE MUST implement AXFR and SHOULD implement IXFR.

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

- Interface Binding: the Hidden Master 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 Master Server is to synchronizes with the Public Authoritative Name Server Set, not to serve zone. As a result, exchanges are performed with specific nodes (the Public Authoritative Servers). Then exchange types are limited. The only legitimate exchanges are: NOTIFY initiated by the Hidden Master and IXFR or AXFR exchanges initiated by the Public Authoritative Name Server Set. On the other hand regular authoritative servers would respond any hosts on the home network, and any DNS(SEC) query would be considered. The CPE SHOULD filter IXFR/AXFR traffic and drop traffic not initiated by the Public Authoritative Server. The CPE MUST listen for DNS on TCP and UDP and at least allow SOA lookups to the DNS Homenet Zone.

5.2. Securing Synchronization

Exchange between the Public Servers and the CPE MUST be secured, at least for integrity protection and for authentication. This is the case whatever mechanism is used between the CPE and the Public Authoritative Name Server Set.

TSIG [RFC2845] or SIG(0) [RFC2931] can be used to secure the DNS communications between the CPE and the Public DNS(SEC) Servers. 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 to roll the keys 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 ease testing and integration. On the other hand, using TSIG, TKEY or SIG(0) requires that these mechanisms to be implemented on the DNS(SEC) Server's implementation running on the CPE, which adds codes. Another disadvantage is that TKEY does not provides authentication mechanism.

Protocols like TLS [RFC5246] / DTLS [RFC6347] can be used to secure the transactions between the Public Authoritative Servers and the CPE. The advantage of TLS/DTLS is that this technology is widely deployed, and most of the boxes already embeds a TLS/DTLS libraries, eventually taking advantage of hardware acceleration. Then TLS/DTLS provides authentication facilities and can use certificates to authenticate the Public Authoritative Server and the CPE. On the other hand, using TLS/DTLS requires to integrate DNS exchange over TLS/DTLS, as well as a new service port. This is why we do not recommend this option.

IPsec [RFC4301] IKEv2 [RFC7296] can also be used to secure the transactions between the CPE and the Public Authoritative Servers. Similarly to TLS/DTLS, most CPE already embeds a IPsec stack, and IKEv2 provides multiple authentications possibilities with its EAP framework. In addition, IPsec can be used to protect the DNS exchanges between the CPE and the Public Authoritative Servers without any modifications of the DNS Servers or client. DNS integration over IPsec only requires an additional security policy in the Security Policy Database. One disadvantage of IPsec is that it hardly goes through NATs and firewalls. However, in our case, the CPE is connected to the Internet, and IPsec communication between the CPE and Public Authoritative Server SHOULD NOT be impacted by middle boxes.

As mentioned above, TSIG, IPsec and TLS/DTLS may be used to secure transactions between the CPE and the Public Authentication Servers. The CPE and Public Authoritative Server SHOULD implement TSIG and IPsec.

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 CPE MAY use PSK or certificate base authentication and that Public Authentication Servers MUST support PSK and certificate based authentication.

5.3. CPE Security Policies

This section details security policies related to the Hidden Master / Slave synchronization.

The Hidden Master, as described in this document SHOULD drop any queries from the home network. This can be performed with port binding and/or firewall rules.

The Hidden Master SHOULD drop on the WAN interface any DNS queries that is not issued from the Public Authoritative Server Name Server Set.

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

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

The Hidden Master SHOULD drop any non protected IXFR or AXFR exchange. This depends how the synchronization is secured.

6. DNSSEC compliant Homenet Architecture

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

Deploying DNSSEC requires signing the zone and configuring a secure delegation. As described in Section 4.1, signing can be performed by the CPE or by the Public Authoritative Servers. Section 6.1 details the implications of these two alternatives. Similarly, the secure delegation can be performed by the CPE or by the Public Authoritative Servers. 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 Public Authoritative Servers. It is recommended to sign the zone by the CPE unless there is a strong argument against it, like a CPE that is not able to sign the zone. In that case zone signing may be performed by the Public Authoritative Servers on behalf of the CPE.

Reasons for signing the zone by the CPE are:

- 1: Keeping the Homenet Zone and the Public Zone equals 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 Master. This could be useful for example if, in the future, the CPE could announce to the home network that the CPE can act a a local authoritative master 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 Zone. On the other hand most CPE 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 Master. Note that outsourcing the zone signing operation requires that all DNSSEC queries be cached to perform a local lookup, otherwise a resolution with the Public Authoritative Master is performed.

- 2: Keeping the Homenet Zone and the Public Zone equals to securely address the connectivity disruption independence exposed in [RFC7368] section 4.4.1 and 3.7.5. As local lookup is 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 a fall back to DNS resolution occurs as the local Homenet Zone is not signed.

- 3: Keeping the Homenet Zone and the Public Zone equals to guarantee coherence between DNS(SEC) 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 DNS Zone are better guaranteed. When the Zone is signed by the CPE, it makes modification of the DNS data -- for example for flow redirection -- not possible. As a result, signing the Homenet Zone by the CPE provides better protection for the end user privacy.

Reasons for signing the zone by the Public Authoritative Servers are:

- 1: The CPE is not able to sign the zone, most likely because its firmware does not make it possible. However the reason is expected to be less and less valid over time.

- 2: Outsourcing DNSSEC management operations. Management operations involve key-roll over which can be done automatically by the CPE and transparently for the end user. As result avoiding DNSSEC management is mostly motivated by bad software implementations.

- 3: Reducing the impact of CPE replacement on the Public Zone. Unless the CPE private keys are backuped, CPE replacement results in a emergency key roll over. This can be mitigated also by using relatively small TTLs.

- 4: Reducing configuration impacts on the end user. Unless there are some zero configuration mechanisms to provide credentials between the new CPE and the Public Authoritative Name Server Sets. Authentications to Public Authoritative Name Server Set should 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: Public Authoritative Servers are more likely to handle securely private keys than the CPE. However, having all private information at one place may also balance that risk.

6.2. Secure Delegation

The secure delegation is set if the DS RRset is properly set in the parent zone. Secure delegation can be performed by the CPE or the Public Authoritative Servers.

The DS RRset can be updated manually by the CPE or the Public Authoritative Servers. This can be used then with nsupdate for example bu requires the CPE or the Public Authoritative Server to be authenticated by the Parent Zone Server. Such a trust channel between the CPE and the Parent Zone server may be hard to maintain, and thus may be easier to establish with the Public Authoritative Server. On the other hand, [RFC7344] may mitigate such issues.

7. Handling Different Views

The DNS Homenet Zone provides information about the home network and some user may be tempted to have different information regarding 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 wild Internet. Each view can be associated to a dedicated Homenet Zone. Note that this document does not specify how DNS queries coming from the home

network are addressed to the DNS(SEC) Homenet Zone. This could be done via the DNS resolver hosted on the CPE for example.

This section is not normative. Section 7.1 expose different reasons that result in different views, Section 7.2 briefly describes the consequences of having distinct views, and Section 7.3 provides guidance to avoid this situation.

7.1. Motivations

The main motivation to handle different views is to provide different information depending on the location the DNS query is emitted. Here are a few motivations for doing so:

- 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 administrate your CPE may not be published on the Internet. Similarly services like the mapper that registers the devices of your home network may not be published on the Internet. In both case, these services should only be known/used by the network administrator. To restrict the access of such services, the home network administrator may chose to publish these information only within the home network, where it may suppose users are more trustable then on the Internet. Even though, this assumption may not be valid, at least, this reduces the surface of 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 Internet.

- 3: If the CPE does not sign the Homenet Zone and outsource the signing process, the two views are at least different since, one is protected with DNSSEC whereas the other is not.

7.2. Consequences

Enabling different views leads to a non-coherent naming system. Basically, depending on where you are 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.

Regarding local-scope IP addresses, such device may end up with poor connectivity. Suppose, for example, the DNS resolution is performed via the WLAN interface attached to the CPE, the response provides local-scope IP addresses and 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, devices with multiple interfaces will have difficulties to secure the naming resolution as responses emitted 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 also happen if the device is able to perform DNS resolutions. DNS resolutions performed via the CPE resolver may be different then those performed over the Internet.

7.3. Guidance and Recommendations

As exposed in Section 7.2, it is recommended to avoid different views. If network administrators chose to implement multiple views, impacts on devices' resolution should be evaluated.

A consequence the DNS(SEC) Homenet Zone is expected to be the exact copy of the DNS(SEC) Public Zone. As a result, services that are not expected to be published on the Internet should not be part of the DNS(SEC) Homenet Zone, local-scope address should not be part of the DNS(SE) Homenet Zone, and when possible, the CPE should sign the DNSSEC Homenet Zone.

The DNS(SEC) Homenet Zone is expected to host public information. It is not to the DNS service to define local home networks 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 one 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 [I-D.ietf-dnssd-requirements].

8. Reverse Zone

Most of the description considered the DNS Homenet Zone as the non-Reverse Zone. This section is focused on the Reverse Zone.

First, all considerations for the DNS Homenet Zone apply to the Reverse Homenet Zone. The main difference between the Reverse DNS Homenet Zone and the DNS Homenet Zone is that the parent zone of the Reverse Homenet 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. If the Reverse Public Authoritative Name Server Set 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 address is so large, that reverse zone may be confronted to a scalability issue. How to reverse zone is generated is out of scope of this document.
[I-D.howard-dnsop-ip6rdns] provides guidance on how to address the scalability issue.

9. Privacy Considerations

Outsourcing the DNS Authoritative service from the CPE to a third entity comes with a a few privacy related concerns.

First the DNS 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 remains accessible though the Internet. Even though DNS makes information public, the DNS does not expect to make the complete list of service 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 of the services. More specifically, making mywebsite.example.com public in the DNS, is not sufficient to make resolvers aware of the existence web site.

In order to prevent the complete DN(SEC) Homenet Zone to be published on the Internet, one should prevent AXFR queries on the Public Authoritative Masters. Similarly, to avoid zone-walking one should prefer NSEC3 [RFC5155] over NSEC [RFC4034].

When the DNS Homenet Zone is outsourced the end user must 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 eventually the device, by as the DNS Homenet Zone contains the IP addresses associated to the service, they limit the scope of the scan.

In addition to the DNS Homenet Zone, the third party can also monitor the traffic associated to the DNS Homenet Zone. This traffic may

provide indication of the services you use, how and when you use these services. Although, cache may alter this information inside the home network, it is likely that outside your home network this information will not be cached.

10. Security Considerations

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

10.1. Names are less secure than IP addresses

This document describes how an End User can make his services and devices from his home network reachable on the Internet with Names rather than IP addresses. This exposes the home network to attackers since names are expected to provide less randomness than IP addresses. The naming delegation protects the End User's privacy by not providing the complete zone of the home network to the ISP. However, using the DNS with names for the home network exposes the home network and its components to dictionary attacks. In fact, with IP addresses, the Interface Identifier is 64 bit length leading to 2^64 possibilities for a given subnetwork. This is not to mention that the subnet prefix is also of 64 bit length, thus providing another 2^64 possibilities. On the other hand, names used either for the home network domain or for the devices present less randomness (livebox, router, printer, nicolas, jennifer, ...) and thus exposes the devices to dictionary attacks.

10.2. 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 the same Prefix over time. As a result observing IP addresses 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 that logging IP addresses, especially to profile 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 we recommend that End Users may choose to respond or not to PTR DNS queries and may return a NXDOMAIN response.

11. IANA Considerations

This document has no actions for IANA.

12. 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 and Michael Richardson, Ray Bellis for their feed backs on handling different views as well as clarifying the impact of outsourcing the zone signing operation outside the CPE.

13. References

13.1. Normative References

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

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

[RFC1995] Ohta, M., "Incremental Zone Transfer in DNS", RFC 1995, August 1996.

[RFC1996] Vixie, P., "A Mechanism for Prompt Notification of Zone Changes (DNS NOTIFY)", RFC 1996, August 1996.

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

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

[RFC2142] Crocker, D., "MAILBOX NAMES FOR COMMON SERVICES, ROLES AND FUNCTIONS", RFC 2142, May 1997.

[RFC2308] Andrews, M., "Negative Caching of DNS Queries (DNS NCACHE)", RFC 2308, March 1998.

[RFC2845] Vixie, P., Gudmundsson, O., Eastlake, D., and B. Wellington, "Secret Key Transaction Authentication for DNS (TSIG)", RFC 2845, May 2000.

[RFC2930] Eastlake, D., "Secret Key Establishment for DNS (TKEY RR)", RFC 2930, September 2000.

[RFC2931] Eastlake, D., "DNS Request and Transaction Signatures (SIG(0)s)", RFC 2931, September 2000.

[RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, March 2005.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RFC5155] Laurie, B., Sisson, G., Arends, R., and D. Blacka, "DNS Security (DNSSEC) Hashed Authenticated Denial of Existence", RFC 5155, March 2008.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC5936] Lewis, E. and A. Hoenes, "DNS Zone Transfer Protocol (AXFR)", RFC 5936, June 2010.

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

[RFC6644] Evans, D., Droms, R., and S. Jiang, "Rebind Capability in DHCPv6 Reconfigure Messages", RFC 6644, July 2012.

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

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

[RFC7296] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., and T. Kivinen, "Internet Key Exchange Protocol Version 2 (IKEv2)", STD 79, RFC 7296, October 2014.

13.2. 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-dnssd-requirements]
Lynn, K., Cheshire, S., Blanchet, M., and D. Migault, "Requirements for Scalable DNS-SD/mDNS Extensions", draft-ietf-dnssd-requirements-04 (work in progress), October 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-00 (work in progress), September 2014.

[RFC1033] Lottor, M., "Domain administrators operations guide", RFC 1033, November 1987.

[RFC7344] Kumari, W., Gudmundsson, O., and G. Barwood, "Automating DNSSEC Delegation Trust Maintenance", RFC 7344, September 2014.

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

Appendix A. Document Change Log

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

-05:

*Clarifying on handling different views:

- 1: How the CPE may be involved in the resolution and responds without necessarily requesting the Public Masters (and eventually the Hidden Master)

- 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 Master(s)". "Public Authoritative Server Management Interface" is replaced by "Public Authoritative Name Server Set".

-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 Master / Slave functions with CPE as Hidden Master/Homenet Server.

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

-00: First version published.

Authors' Addresses

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

Email: mglt.ietf@gmail.com

Wouter Cloetens
SoftAtHome
vaartdijk 3 701
3018 Wijgmaal
Belgium

Email: wouter.cloetens@softathome.com

Chris Griffiths
Dyn
150 Dow Street
Manchester, NH 03101
US

Email: cgriffiths@dyn.com
URI: http://dyn.com

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

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

Homenet Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 6, 2015

M. Stenberg

S. Barth

P. Pfister
Cisco Systems
March 5, 2015

Home Networking Control Protocol
draft-ietf-homenet-hncp-04

Abstract

This document describes the Home Networking Control Protocol (HNCP), an extensible configuration protocol and a set of requirements for home network devices on top of the Distributed Node Consensus Protocol (DNCP). It enables automated configuration of addresses, naming, network borders and the seamless use of a routing protocol.

Table of Contents

1. Introduction

HNCP synchronizes state across a small site in order to allow automated network configuration. The protocol enables use of border discovery, address prefix distribution [I-D.ietf-homenet-prefix-assignment], naming and other services across multiple links.

HNCP provides enough information for a routing protocol to operate without homenet-specific extensions. In homenet environments where multiple IPv6 source-prefixes can be present, routing based on source and destination address is necessary [RFC7368].

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 RFC 2119 [RFC2119].

3. DNCP Profile

HNCP is defined as a profile of DNCP [I-D.ietf-homenet-dncp] with the following parameters:

- HNCP uses UDP datagrams on port HNCP-UDP-PORT as a transport over link-local scoped IPv6, using unicast and multicast (group All-Homenet-Routers). Received datagrams with an IPv6 source or destination address which is not link-local scoped MUST be ignored. IPv6 fragmentation and reassembly up to TBD bytes MUST be supported by the stack of the node.

- HNCP operates on multicast-capable interfaces only, thus every DNCP Connection Identifier MUST refer to one, except for the value 0 which is reserved for internal purposes and MUST NOT be used for connection enumeration. Implementations MAY use a value equivalent to the sin6_scope_id for the given interface.

- HNCP unicast messages SHOULD be secured using DTLS [RFC6347] as described in DNCP if exchanged over unsecured links. UDP on port HNCP-DTLS-PORT is used for this purpose. A node implementing the security mechanism MUST support the DNCP Pre-Shared Key method, SHOULD support the DNCP Certificate Based Trust Consensus and MAY support the PKI-based trust method.

- HNCP uses opaque 32-bit node identifiers (DNCP_NODE_IDENTIFIER_LENGTH = 32). A node implementing HNCP SHOULD generate and use a random node identifier. If it receives a Node State TLV with the same node identifier and a higher update sequence number, it MUST immediately generate and use a new random node identifier which is not used by any other node.

- HNCP nodes MUST treat all Long Network State Update messages received via multicast on a link which has DNCP security enabled as if they were Short Network State Update messages, i.e. they MUST ignore all contained Node State TLVs.

- HNCP nodes use the following Trickle parameters:

  * k SHOULD be 1, given the timer reset on data updates and retransmissions should handle packet loss.

  * 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 (i.e. 25.6 seconds) but MUST NOT be lower.

- HNCP nodes MUST use the leading 64 bits of MD5 [RFC1321] as DNCP non-cryptographic hash function H(x).

- HNCP nodes MUST use the keep-alive extension on all connections. The default keep-alive interval (DNCP_KEEPALIVE_INTERVAL) is 20 seconds, the multiplier (DNCP_KEEPALIVE_MULTIPLIER) MUST be 2.1, the grace-interval (DNCP_GRACE_INTERVAL) SHOULD be equal to DNCP_KEEPALIVE_MULTIPLIER times DNCP_KEEPALIVE_INTERVAL.

4. Adjacent Links

HNCP uses the concept of Adjacent Links for some of its applications. This term is defined as follows:

If the connection of a node is detected or configured to be an ad-hoc interface the Adjacent Link only consists of said interface.

Otherwise the Adjacent Link contains all interfaces bidirectionally reachable from a given local interface. An interface X of a node A and an interface Y of a node B are bidirectionally reachable if and only if node A publishes a Neighbor TLV with the Neighbor Node Identifier B, the Neighbor Connection Identifier Y and the Local Connection Identifier X and node B publishes a Neighbor TLV with the Neighbor Node Identifier A, a Neighbor Connection Identifier X and the Local Connection Identifier Y. In addition a node MUST be able

to detect whether two of its local interfaces belong to the same Adjacent Link either by local means or by inferring that from the bidirectional reachability between two different local interfaces and the same remote interface.

5. Border Discovery

HNCP associates each HNCP interface with a category (e.g., internal or external). This section defines the border discovery algorithm derived from the edge router interactions described in the Basic Requirements for IPv6 Customer Edge Routers [RFC7084]. This algorithm is suitable for both IPv4 and IPv6 (single or dual-stack) and determines whether an HNCP interface is internal, external, or uses another fixed category. This algorithm MUST be implemented by any router implementing HNCP.

In order to avoid conflicts between border discovery and homenet routers running DHCPv4 [RFC2131] or DHCPv6-PD [RFC3633] servers, each router MUST implement the following mechanism based on The User Class Option for DHCPv4 [RFC3004] and its DHCPv6 counterpart [RFC3315]:

- An HNCP router running a DHCP client on a homenet interface MUST include a DHCP User-Class consisting of the ASCII-String "HOMENET".

- An HNCP router running a DHCP server on a homenet interface MUST ignore or reject DHCP-Requests containing a DHCP User-Class consisting of the ASCII-String "HOMENET".

The border discovery auto-detection algorithm works as follows, with evaluation stopping at first match:

1. If a fixed category is configured for the interface, it MUST be used.

2. If a delegated prefix could be acquired by running a DHCPv6 client on the interface, it MUST be considered external.

3. If an IPv4 address could be acquired by running a DHCPv4 client on the interface it MUST be considered external.

4. Otherwise the interface MUST be considered internal.

A router MUST allow setting a category of either auto-detected, internal or external for each interface which is suitable for both internal and external connections. In addition the following specializations of the internal category are defined to modify the local router behavior:

Leaf category: This declares an interface used by client devices only. A router MUST consider such interface as internal but MUST NOT send nor receive HNCP messages on such interface. A router SHOULD implement this category.

Guest category: This declares an interface used by untrusted client devices only. In addition to the restrictions of the Leaf category, connected devices MUST NOT be able to reach other devices inside the HNCP network nor query services advertised by them unless explicitly allowed, instead they SHOULD only be able to reach the internet. This category SHOULD be supported.

Ad-hoc category: This configures an interface to be ad-hoc (Section 4) and MAY be implemented.

Hybrid category: This declares an interface to be internal while still using external connections on it. It is assumed that the link is under control of a legacy, trustworthy non-HNCP router, still within the same network. Detection of this category automatically in addition to manual configuration is out of scope for this document. This category MAY be implemented.

Each router MUST continuously scan each active interface that does not have a fixed category in order to dynamically reclassify it if necessary. The router therefore runs an appropriately configured DHCPv4 and DHCPv6 client as long as the interface is active including states where it considers the interface to be internal. 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. Once it treats a certain interface as internal it MUST start forwarding traffic with appropriate source addresses between its internal interfaces and allow internal traffic to reach external networks according to the routes it publishes. Once a router detects an interface transitioning to external it MUST stop any previously enabled internal forwarding. In addition it SHOULD announce the acquired information for use in the network as described in later sections of this draft if the interface appears to be connected to an external network.

6. Autonomic Address Configuration

This section specifies how HNCP routers configure host and router addresses. At first border routers share information obtained from service providers or local configuration by publishing one or more External Connection TLVs. These contain other TLVs such as Delegated Prefix TLVs which are then used for prefix assignment. Finally, HNCP routers obtain addresses using a stateless (SLAAC-like) procedure or

a specific stateful mechanism and hosts and legacy routers are configured using SLAAC or DHCP.

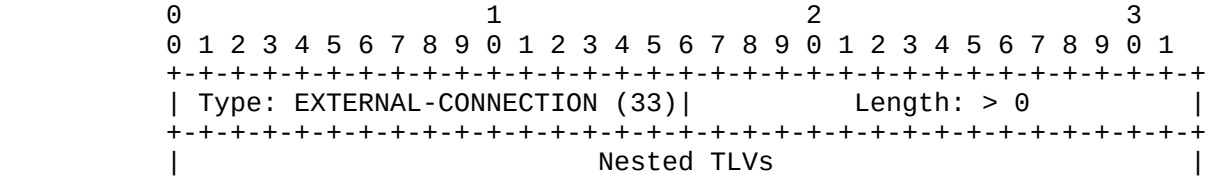
In all TLVs specified in this section which include a prefix, IPv4 prefixes are encoded using the IPv4-mapped IPv6 addresses format [RFC4291]. The prefix length of such prefix is set to 96 plus the IPv4 prefix length.

6.1. External Connections

Each HNCP router MAY obtain external connection information from one or more sources, e.g. DHCPv6-PD [RFC3633], NETCONF [RFC6241] or static configuration. This section specifies how such information is encoded and advertised.

6.1.1. External Connection TLV

An External Connection TLV is a container-TLV used to gather network configuration information associated with a single external connection. A node MAY publish zero, one or more instances of this TLV.



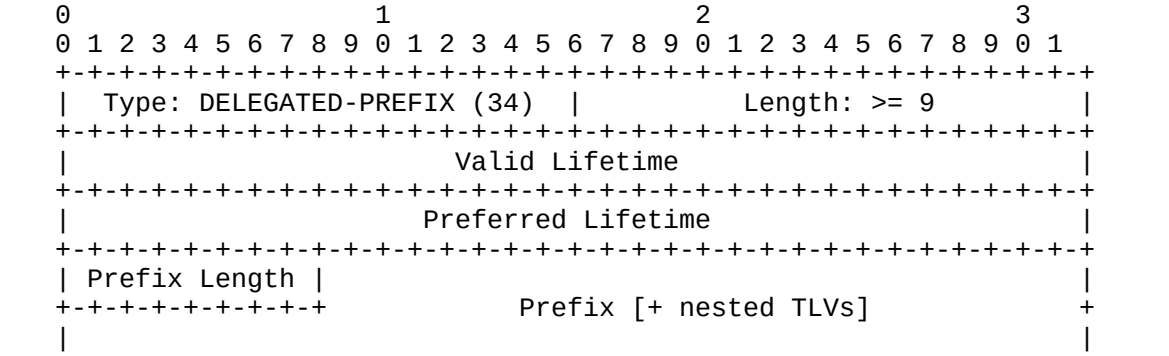
The External Connection TLV is a container which:

- MAY contain zero, one or more Delegated Prefix TLVs.
- MUST NOT contain multiple Delegated Prefix TLVs with the same prefix. In such a situation, the container MUST be ignored.
- MAY contain at most one DHCPv6 Data TLV and at most one DHCPv4 Data TLV encoding options associated with the External Connection but MUST NOT contain more than one of each otherwise the whole External Connection TLV MUST be ignored.
- MAY contain other TLVs for future use.

6.1.2. Delegated Prefix TLV

The Delegated Prefix TLV is used by HNCP routers to advertise prefixes which are allocated to the whole network and will be used

for prefix assignment. All Delegated Prefix TLVs MUST be nested in an External Connection TLV.



Valid Lifetime: The time in seconds the delegated prefix is valid. The value is relative to the point in time the Node-Data TLV was last published. It MUST be updated whenever the node republishes its Node-Data TLV.

Preferred Lifetime: The time in seconds the delegated prefix is preferred. The value is relative to the point in time the Node-Data TLV was last published. It MUST be updated whenever the node republishes its Node-Data TLV.

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

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

Nested TLVs: Other TLVs included in the Delegated Prefix TLV and starting at the next 32 bits boundary following the end of the encoded prefix.

If the encoded prefix represents an IPv6 prefix, at most one DHCPv6 Data TLV MAY be included.

If the encoded prefix represents an IPv4-mapped IPv6 address, at most one DHCPv4 Data TLV MAY be included.

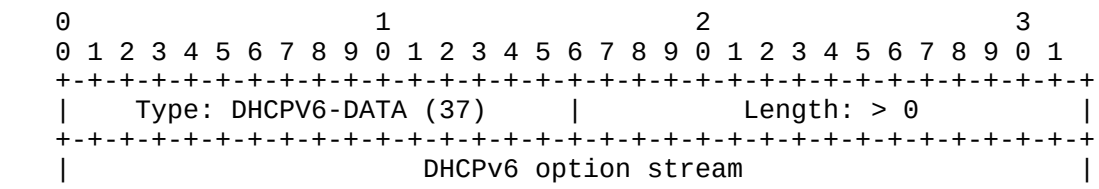
It MAY contain other TLVs for future use.

6.1.3. DHCP Data TLVs

Auxiliary connectivity information is encoded as a stream of DHCP options. Such TLVs MUST only be present in an External Connection TLV or a Delegated Prefix TLV. When included in an External

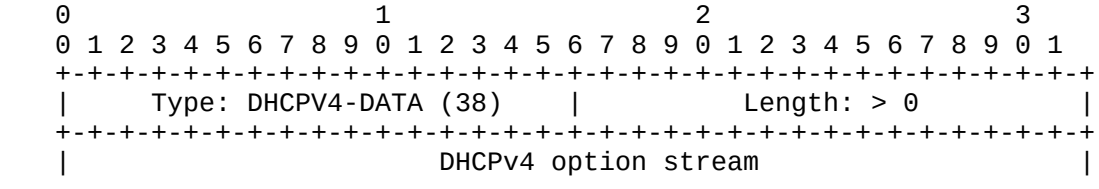
Connection TLV, they MUST contain DHCP options which are relevant to the whole External Connection. When included in a Delegated Prefix, they MUST contain DHCP options which are specific to the Delegated Prefix.

The DHCPv6 Data TLV uses the following format:



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

The DHCPv4 Data TLV uses the following format:



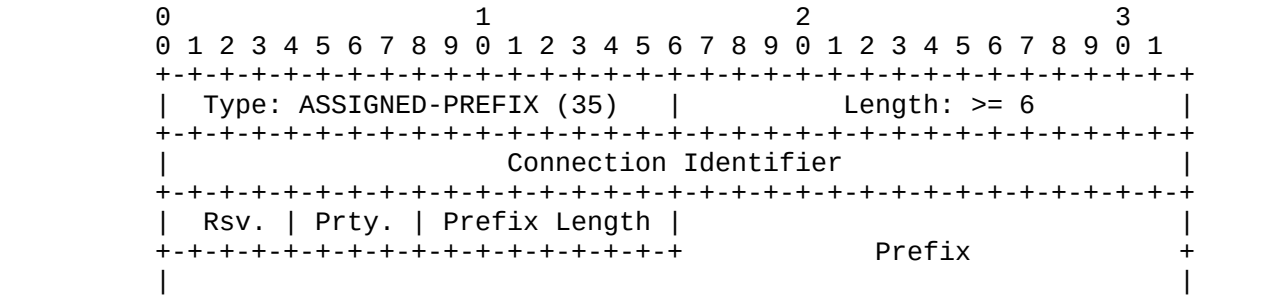
DHCPv4 option stream: DHCPv4 options encoded as specified in [RFC2131].

6.2. Prefix Assignment

HNCP uses the Distributed Prefix Assignment Algorithm specified in [I-D.ietf-homenet-prefix-assignment] in order to assign prefixes to HNCP internal links and uses the terminology defined there.

6.2.1. Assigned Prefix TLV

Published Assigned Prefixes MUST be advertised using the Assigned Prefix TLV:



Connection Identifier: The DNCP Connection Identifier of the link the prefix is assigned to, or 0 if the link is a Private Link.

Rsv.: Bits reserved for future use. MUST be set to zero when creating this TLV and ignored when parsing it.

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 specified in the router’s configuration.
- 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 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 zeroes up to the next byte boundary.

6.2.2. Prefix Assignment Algorithm Parameters

All HNCP nodes running the prefix assignment algorithm MUST use the following parameters:

Node IDs: DNCP Node Identifiers are used. The comparison operation is defined as bit-wise comparison.

Set of Delegated Prefixes: The set of prefixes encoded in Delegated Prefix TLVs which 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 HNCP internal, leaf, guest or hybrid links. It is dynamically updated as HNCP links are added, removed, become internal or cease to be.

Set of Private Links: This document defines Private Links representing DHCPv6-PD clients or as a mean to advertise prefixes included in the DHCPv6 Exclude Prefix option. Other implementation-specific Private Links may exist.

Set of Advertised Prefixes: The set of prefixes included in Assigned Prefix TLVs advertised by other HNCP routers. 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 zero, the Advertised Prefix is not assigned on a Shared Link.

* If the Link Identifier is not zero the Shared Link is equal to the Adjacent Link (Section 4). Advertised Prefixes as well as their associated priorities and associated Shared Links MUST be updated as Assigned Prefix TLVs or Neighbor TLVs are added, removed or updated.

ADOPT_MAX_DELAY: The default value is 0 seconds (i.e. prefix adoption MAY be 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.

6.2.3. Making New Assignments

Whenever the Prefix Assignment Algorithm routine is run on an Adjacent Link and whenever a new prefix may be assigned (case 1 of the routine), the decision of whether the assignment of a new prefix is desired MUST follow these rules:

- If the Delegated Prefix TLV contains a DHCPv4 or DHCPv6 Data TLV, and the meaning of one of the DHCP options is not understood by the HNCP router, the creation of a new prefix is not desired.

- 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-null preferred lifetime, the creation of a new prefix is not desired.

- Otherwise, the creation of a new prefix is 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 by an administrator. In case no prefix of length 64 would be available, a longer prefix MAY be selected.

If the considered delegated prefix is an IPv4 prefix (Section 6.4 details how IPv4 delegated prefixes are generated), a prefix of length 24 SHOULD be preferred.

In any case, a router MUST support a mechanism suitable to distribute addresses from the considered prefix to clients on the link. Otherwise it MUST NOT create or adopt it, i.e. a router assigning an IPv4 prefix MUST support the L-capability and a router assigning an IPv6 prefix not suitable for stateless autoconfiguration MUST support the H-capability as defined in Section 10.

6.2.4. Applying Assignments

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

- MUST create an appropriate on-link route for said prefix and advertise it using the chosen routing protocol.

- MUST participate in the client configuration election as described in Section 7.

MAY add an address from said prefix to the respective network interface as described in Section 6.3.

6.2.5. DHCPv6-PD Excluded Prefix Support

Whenever a DHCPv6 Prefix Exclude option [RFC6603] is received with a delegated prefix, the excluded prefix MUST be advertised as assigned to a Private Link with the maximum priority (i.e. 15).

The same procedure MAY be applied in order to exclude prefixes obtained by other means of configuration.

6.2.6. Downstream Prefix Delegation Support

When an HNCP router receives a request for prefix delegation, it SHOULD assign one prefix per delegated prefix, wait for them to be applied, and delegate them to the client. Such assignment MUST be done in accordance with the Prefix Assignment Algorithm. Each client MUST be considered as an independent Private Link and delegation MUST be based on the same set of Delegated Prefixes.

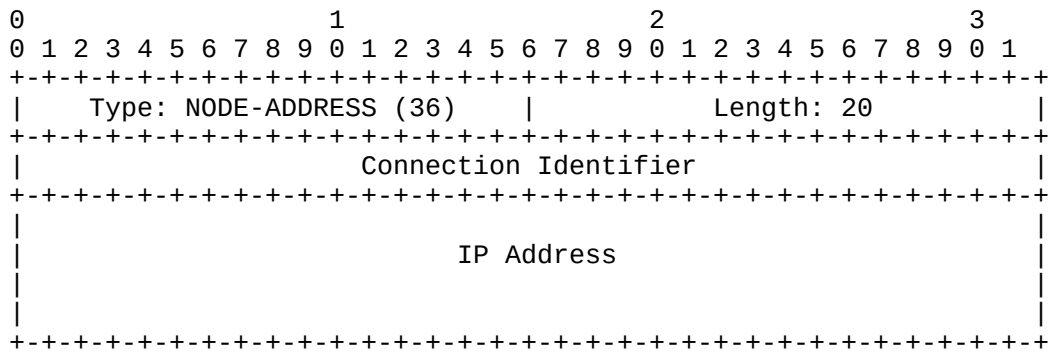
The assigned prefixes MUST NOT be given to clients before they are applied, and MUST be withdrawn whenever they are destroyed. As an exception to this rule a router MAY prematurely give out a prefix which is advertised but not yet applied if it does so with a valid lifetime of not more than 30 seconds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

6.3. 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. SLAAC SHOULD be used whenever possible. The following method MUST be used otherwise.

For any IPv6 prefix longer than 64 bits (resp. any IPv4 prefix) assigned to an Adjacent Link, the first quarter of the addresses are reserved for routers HNCP based assignments, whereas the last three quarters are left to the DHCPv6 (resp. DHCPv4) elected router (Section 10 specifies the DHCP server election process). For instance, if the prefix 192.0.2.0/24 is assigned and applied to an Adjacent 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 routers assign themselves addresses using the Node Address TLV:



Connection Identifier: The DNCP Connection Identifier of the link the address is assigned to, or 0 if it is not assigned on an HNCP enabled link.

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

The process of obtaining addresses is specified as follows:

- A router MUST NOT start advertising an address if it is already advertised by another router.
- An assigned address MUST be in the first quarter of an assigned prefix currently applied on the specified link.
- 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 highest node identifier MUST be removed.

6.4. Local IPv4 and ULA Prefixes

HNCP routers can create an ULA or private IPv4 prefix to enable connectivity between local devices. These prefixes are inserted in HNCP as if they were delegated prefixes. The following rules apply:

An HNCP router SHOULD create an ULA prefix if there is no other non-deprecated IPv6 prefix in the network. It MAY also do so if there are other delegated IPv6 prefixes but none of which is a non-deprecated ULA but MUST NOT do so otherwise. Whenever it detects another non-deprecated ULA being advertised it MUST cease

to announce its locally generated one. It MAY also do so once it detects a non-deprecated non-ULA IPv6 delegated prefix.

An HNCP router MUST create a non-deprecated private IPv4 prefix [RFC1918] whenever it wishes to provide external IPv4 connectivity to the network and no other non-deprecated private IPv4 prefix currently exists. It MAY also create a deprecated private IPv4 prefix if no IPv4 prefix exists and it wants to enable local IPv4 connectivity but MUST NOT do so otherwise. In case multiple IPv4 prefixes are announced all but one MUST be removed while non-deprecated ones take precedence over deprecated ones and announcements by nodes with a higher node identifier take precedence over those with a lower one. The router publishing the non-deprecated prefix MUST announce an IPv4 default route using the routing protocol and perform NAT on behalf of the network as long as it publishes the prefix, other routers in the network MUST NOT.

Creation of such ULA and IPv4 prefixes MUST be delayed by a random timespan between 0 and 10 seconds in which the router MUST scan for other nodes 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 an 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 10 in order to participate as a candidate in the election.

7.1. DHCPv6 for Addressing or Configuration

In general stateless address configuration is preferred whenever possible since it enables fast renumbering and low overhead, however stateful DHCPv6 can be useful in addition to collect hostnames and use them to provide naming services or if stateless configuration is not possible for the assigned prefix length.

The designated stateful DHCPv6 server for a link is elected based on the capabilities described in Section 10. The winner is the router

connected to the Adjacent Link (Section 4) advertising the greatest H-capability. In case of a tie, Capability Values and node identifiers are considered (greatest value is elected). The elected router MUST serve stateful DHCPv6 and Router Advertisements on the given link. Furthermore it MUST provide naming services for acquired hostnames as outlined in Section 8. Stateful addresses being handed out SHOULD have a low preferred lifetime (e.g. 1s) to not hinder fast renumbering if either the DHCPv6 server or client do not support the DHCPv6 reconfigure mechanism and the address is from a prefix for which stateless autoconfiguration is supported as well. In case no router was elected, stateful DHCPv6 is not provided and each router assigning IPv6-prefixes on said link MUST provide stateless DHCPv6 service.

7.2. Sending Router Advertisements

Each HNCP router assigning an IPV6-prefix to an interface MUST send Router Advertisements periodically via multicast and via unicast in response to Router Solicitations. In addition other routers on the link MAY announce Router Advertisements. This might result in a more optimal routing decision for clients. The following rules MUST be followed when sending Router Advertisements:

The "Managed address configuration" flag MUST be set whenever a router connected to the link is advertising a non-null H-capability and MUST NOT be set otherwise. The "Other configuration" flag MUST always be set.

The default Router Lifetime MUST be set to an appropriate non-null value whenever an IPv6 default route is known in the HNCP network and MUST be set to zero otherwise.

A Prefix Information Option MUST be added for each assigned and applied IPv6 prefix on the given link. The autonomous address-configuration flag MUST be set whenever the prefix is suitable for stateless configuration. The preferred and valid lifetimes MUST be smaller than the preferred and valid lifetimes of the delegated prefix the prefix is from. When a prefix is removed, it MUST be deprecated as specified in [RFC7084].

A Route Information Option [RFC4191] MUST be added for each delegated IPv6 prefix known in the HNCP network. Additional ones SHOULD be added for each non-default IPv6 route with an external destination advertised by the routing protocol.

A Recursive DNS Server Option and a DNS Search List Option MUST be included with appropriate contents.

To allow for optimized routing decisions for clients on the local link routers SHOULD adjust their Default Router Preference and Route Preferences [RFC4191] so that the priority is set to low if the next hop of the default or more specific route is on the same interface as the Route Advertisement being sent on. Similarly the router MAY use the high priority if it is certain it has the best metric of all routers on the link for all routes known in the network with the respective destination.

Every router sending Router Advertisements MUST immediately send an updated Router Advertisement via multicast as soon as it notices a condition resulting in a change of any advertised information.

7.3. DHCPv6 for Prefix Delegation

The designated DHCPv6 server for prefix-delegation on a link is elected based on the capabilities described in Section 10. The winner is the router connected to the Adjacent Link (Section 4) advertising the greatest P-capability. In case of a tie, Capability Values and Node Identifiers are considered (greatest value is elected). The elected router MUST provide prefix-delegation services [RFC3633] on the given link and follow the rules in Section 6.2.6.

7.4. DHCPv4 for Adressing and Configuration

The designated DHCPv4 server on a link is elected based on the capabilities described in Section 10. The winner is the router connected to the Adjacent Link (Section 4) advertising the greatest L-capability. In case of a tie, Capability Values and node identifiers are considered (greatest value is elected). The elected router MUST provide DHCPv4 services on the given link.

The DHCPv4 serving router MUST announce itself as router [RFC2132] to clients if and only if there is an IPv4 default route known in the network. In addition the router SHOULD announce a Classless Static Route Option [RFC3442] for each non-default IPv4 route advertised in the routing protocol with an external destination.

DHCPv4 lease times SHOULD be short (i.e. not longer than 5 minutes) in order to provide reasonable response times to changes.

7.5. Multicast DNS Proxy

The designated MDNS [RFC6762]-proxy on a link is elected based on the capabilities described in Section 10. The winner is the router with the highest Node Identifier among those with the highest Capability Value on the link that support the M-capability. 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

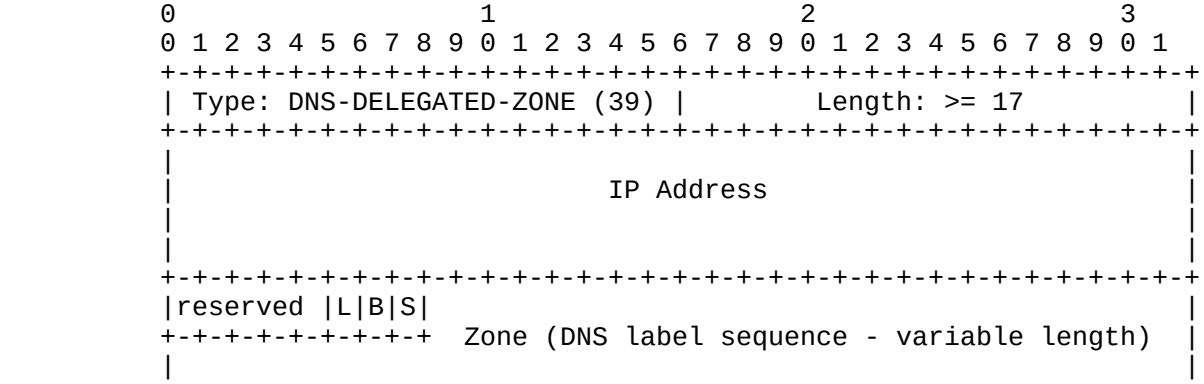
Network-wide naming and service discovery can greatly improve the user-friendliness of an IPv6 network. The following mechanism provides means to setup and delegate naming and service discovery across multiple HNCP routers.

Each HNCP router SHOULD provide and announce an auto-generated or user-configured name for each internal Adjacent Link (Section 4) for which it is the designated DHCPv4, stateful DHCPv6 server or MDNS [RFC6762]-proxy and for which it provides DNS-services on behalf of devices on said link. In addition it MAY provide reverse lookup services.

The following TLVs are defined and MUST be supported by all nodes implementing naming and service discovery:

8.1. 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. There MUST NOT be more than one delegation for the same zone in the whole DNCP network. In case of a conflict the announcement of the node with the highest node identifier takes precedence and all other nodes MUST cease to announce the conflicting TLV.



IP Address is 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-zero) means the delegation is available in the global DNS-hierarchy.

reserved bits MUST be zero when creating and ignored when parsing this TLV.

L-bit (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.

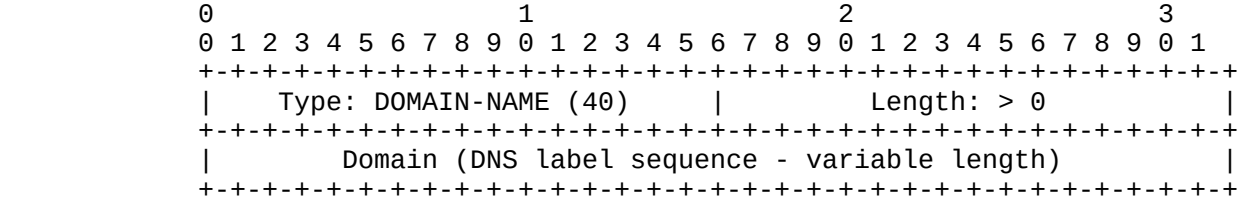
B-bit (DNS-SD [RFC6763] Browse) indicates that this delegated zone should be included in the network's DNS-SD browse list of domains at b._dns-sd._udp. (DOMAIN-NAME). Local forward zones SHOULD have this bit set, reverse zones SHOULD NOT.

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 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 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 is the label sequence of the zone, encoded according to [RFC1035]. Compression MUST NOT be used. The zone MUST end with an empty label.

8.2. Domain Name TLV

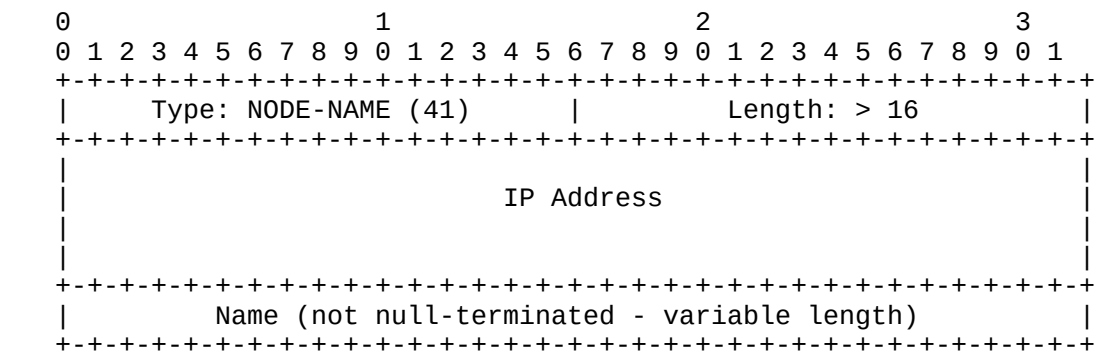
This TLV is used to indicate the base domain name for the network. It is the zone used as a base for all non fully-qualified delegated zones and node names. In case of conflicts the announced domain of the node with the highest node identifier takes precedence. By default ".home" is used, i.e. if no node advertises such a TLV.



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

8.3. Node Name TLV

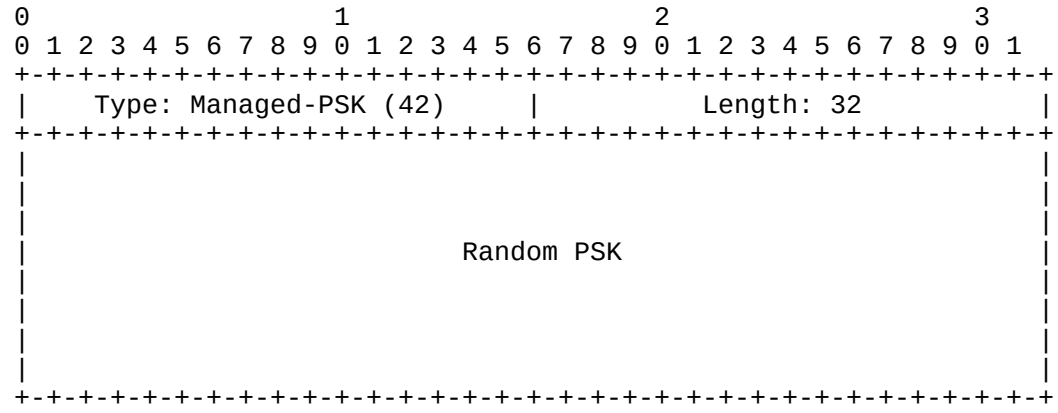
This TLV is used to assign the name of a node in the network to a certain IP address. In case of conflicts the announcement of the node with the highest node identifier for a name takes precedence and all other nodes MUST cease to announce the conflicting TLV.



9. Securing Third-Party Protocols

Pre-shared keys (PSKs) are often required to secure IGPs and other protocols which lack support for asymmetric security. The following mechanism manages PSKs using HNCP to enable bootstrapping of such third-party protocols and SHOULD therefore be used if such a need arises. The following rules define how such a PSK is managed and used:

- If no Managed-PSK-TLV is currently being announced, an HNCP router MUST create one with a 32 bytes long random key and add it to its node data.
- In case multiple routers announce such a TLV at the same time, all but the one with the highest node identifier stop advertising it and adopt the remaining one.
- The router currently advertising the Managed-PSK-TLV must generate and advertise a new random one whenever an unreachable node is purged as described in DNCP.



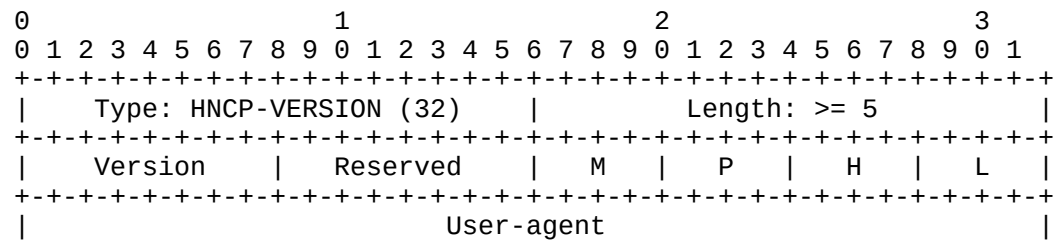
PSKs for individual protocols are derived from the random PSK through the use of HMAC-SHA256 [RFC6234] with a pre-defined per-protocol HMAC-key in ASCII-format. The following HMAC-keys are currently defined to derive PSKs for the respective protocols:

"ROUTING": to be used for IGPs

10. HNCP Versioning and Capabilities

Multiple versions of HNCP based on compatible DNCP [I-D.ietf-homenet-dncp] profiles may be present in the same network when transitioning between HNCP versions and HNCP routers may have different capabilities to support clients. The following mechanism describes a way to announce the currently active version and User-agent of a node. Each node MUST include an HNCP-Version-TLV in its Node Data and MUST ignore (except for DNCP synchronization purposes) any TLVs with a type greater than 32 of nodes not publishing an HNCP-Version TLV or publishing such a TLV with a different Version number.

Capabilities are indicated by setting M, P, H and L fields in the TLV. The "capability value" is a metric indicated by interpreting the bits as an integer, i.e. (M << 12 | P << 8 | H << 4 | L).



Version: Version indicates which version of HNCP is currently in use by this particular node. It MUST be set to 0. Nodes with different versions are considered incompatible.

Reserved: Bits reserved for future use. MUST be set to zero when creating this TLV and ignored when parsing it.

M-capability: Priority value used for electing the on-link MDNS [RFC6762] proxy. It MUST be set to some value between 1 and 7 included (4 is the default) if the router is capable of proxying MDNS and 0 otherwise. 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 some value between 1 and 7 included (4 is the default) if the router is capable of providing prefixes through DHCPv6-PD (Section 6.2.6) and 0 otherwise. 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 some value between 1 and 7 included (4 is the default) if the router is capable of providing such addresses and 0 otherwise. 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 some value between 1 and 7 included (4 is the default) if the router is capable of running a legacy DHCPv4 server offering IPv4 addresses to clients and 0 otherwise. The values 8-15 are reserved for future use.

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

11. Requirements for HNCP Routers

Each router implementing HNCP is subject to the following requirements:

- It MUST implement HNCP-Versioning, Border Discovery, Prefix Assignment and Configuration of hosts and non-HNCP routers as defined in this document.

- It MUST implement and run the method for securing third-party protocols whenever it uses the security mechanism of HNCP.

- It SHOULD implement support for the Service Discovery and Naming TLVs as defined in this document.

- It MUST implement and run the routing protocol $FOO with support for source-specific routes on all of the interfaces it sends and receives HNCP-messages on and MUST resort to announcing source-specific routes for external destinations appropriately.

- It MUST use adequate security mechanisms for the routing protocol on any interface where it also uses the security mechanisms of HNCP. If the security mechanism is based on a PSK it MUST use a PSK derived from the Managed-PSK to secure the IGP.

- It MUST comply with the Basic Requirements for IPv6 Customer Edge Routers [RFC7084] unless it would otherwise conflict with any requirements in this document (e.g. prefix assignment mandating a different prefix delegation and DHCP server election strategy). In general "WAN interface requirements" shall apply to external interfaces and "LAN interface requirements" to internal interfaces respectively.

- It SHOULD be able to provide connectivity to IPv4-devices using DHCPv4.

- It SHOULD be able to delegate prefixes to legacy IPv6 routers using DHCPv6-PD.

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 [I-D.ietf-homenet-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 adjacent links

- o Address assignment information such as delegated and assigned prefixes for individual links

- o Naming and service discovery information such as auto-generated or customized names for individual links and routers

12.1. Border Determination

As described in Section 5, an HNCP router determines the internal or external state on a per-link basis. A firewall perimeter is set up for the external links, and for internal links, HNCP and IGP traffic is allowed.

Threats concerning automatic border discovery 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 adequate 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 which is enabled for HNCP traffic. If left unsecured, attackers may perform arbitrary eavesdropping, spoofing or denial of service attacks on HNCP services such as address assignment or service discovery.

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 and IGP 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, DNCP secure mode 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 is requested to maintain a registry for HNCP TLV-Types.

HNCP inherits the TLV-Types and allocation policy defined in DNCP [I-D.ietf-homenet-dncp]. In addition the following TLV-Types are defined in this document:

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

HNCP requires allocation of UDP port numbers HNCP-UDP-PORT and HNCP-DTLS-PORT, as well as an IPv6 link-local multicast address All-Homenet-Routers.

14. References

14.1. Normative references

[I-D.ietf-homenet-dncp]
Stenberg, M. and S. Barth, "Distributed Node Consensus Protocol", draft-ietf-homenet-dncp-01 (work in progress), March 2015.

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

[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.

[RFC4191] Draves, R. and D. Thaler, "Default Router Preferences and More-Specific Routes", RFC 4191, November 2005.

[I-D.ietf-homenet-prefix-assignment]
Pfister, P., Paterson, B., and J. Arkko, "Distributed Prefix Assignment Algorithm", draft-ietf-homenet-prefix-assignment-03 (work in progress), February 2015.

14.2. Informative references

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

[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.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[RFC3442] Lemon, T., Cheshire, S., and B. Volz, "The Classless Static Route Option for Dynamic Host Configuration Protocol (DHCP) version 4", RFC 3442, December 2002.

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

14.3. URIs

[3] http://www.openwrt.org

[4] http://www.homewrt.org/doku.php?id=run-conf

Appendix A. Some Outstanding Issues

Should we define in-protocol fragmentation scheme or just use IPv6 fragmentation with 'big enough' minimum acceptable size allowed for implementations? In the same sense should we use TCP over UDP?

How should we deal with stub-routers requiring reduced complexity versions? Stub IGPs? An HNCP-based solution? Even a stub-HNCP?

Appendix B. Changelog

draft-ietf-homenet-hncp-03: Split to DNCP (generic protocol) and HNCP (homenet profile).

draft-ietf-homenet-hncp-02: Removed any built-in security. Relying on IPsec. Reorganized interface categories, added requirements languages, made manual border configuration a MUST-support. Redesigned routing protocol election to consider non-router devices.

draft-ietf-homenet-hncp-01: Added (MAY) guest, ad-hoc, hybrid categories for interfaces. Removed old hnetv2 reference, and now pointing just to OpenWrt + github. Fixed synchronization algorithm to spread also same update number, but different data hash case. Made purge step require bidirectional connectivity between nodes when traversing the graph. Edited few other things to be hopefully slightly clearer without changing their meaning.

draft-ietf-homenet-hncp-00: Added version TLV to allow for TLV content changes pre-RFC without changing IDs. Added link id to assigned address TLV.

Appendix C. Draft source

This draft is available at https://github.com/fingon/ietf-drafts/ in source format. Issues and pull requests are welcome.

Appendix D. Implementation

A GPLv2-licensed implementation of HNCP is currently under development at https://github.com/sbyx/hnetd/ and binaries are available in the OpenWrt [3] package repositories. See [4] for more information. Feedback and contributions are welcome.

Appendix E. Acknowledgements

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

Thanks to Eric Kline for the original border discovery work.

Authors' Addresses

Markus Stenberg
Helsinki 00930
Finland

Email: markus.stenberg@iki.fi

Steven Barth
Halle 06114
Germany

Email: cyrus@openwrt.org

Pierre Pfister
Cisco Systems
Paris
France

Email: pierre.pfister@darou.fr

Homenet Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 6, 2015

M. Stenberg

March 5, 2015

Auto-Configuration of a Network of Hybrid Unicast/Multicast DNS-Based Service Discovery Proxy Nodes
draft-ietf-homenet-hybrid-proxy-zeroconf-00

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 September 6, 2015.

Table of Contents

1. Introduction

Section 3 ("Hybrid Proxy Operation") of [I-D.cheshire-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.cheshire-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. 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).

7. IANA Considerations

This document has no actions for IANA.

8. References

8.1. Normative references

[I-D.cheshire-dnssd-hybrid]
Cheshire, S., "Hybrid Unicast/Multicast DNS-Based Service Discovery", draft-cheshire-dnssd-hybrid-01 (work in progress), January 2014.

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

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

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

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

8.2. Informative references

[I-D.ietf-homenet-hncp]
Stenberg, M., Barth, S., and P. Pfister, "Home Networking Control Protocol", draft-ietf-homenet-hncp-03 (work in progress), January 2015.

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

[RFC3646] Droms, R., "DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3646, December 2003.

8.3. 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.

Author’s Address

Markus Stenberg
Helsinki 00930
Finland

Email: markus.stenberg@iki.fi

HOMENET
Internet-Draft
Intended status: Standards Track
Expires: August 21, 2015

D. Migault (Ed)
Ericsson
W. Cloetens
SoftAtHome
C. Griffiths
Dyn
R. Weber
Nominum

February 17, 2015

DHCP Options for Homenet Naming Architecture
draft-ietf-homenet-naming-architecture-dhc-options-01.txt

Abstract

CPEs are usually constraint 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. One way to avoid exposing CPE is to outsource the authoritative service to a third party. 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 which may be unappropriated for most end users. To leverage this issue, this document proposes DHCP Options so any agnostic CPE can automatically proceed to the appropriated configuration and outsource the authoritative naming service for the home network. This document shows that in most cases, these DHCP Options make outsourcing to a third party (be it the ISP or any ISP independent service provider) transparent for the end user.

This Internet-Draft will expire on August 21, 2015.

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. Terminology

- Customer Premises Equipment: (CPE) is the router providing connectivity to the home network. It is configured and managed by the end user. In this document, the CPE might also hosts services such as DHCPv6. This device might be provided by the ISP.

- Public Key: designates a public Key generated by the CPE. This key is used as an authentication credential for the CPE.

- Registered Homenet Domain: is the Domain Name associated to the home network.

- DNS Homenet Zone: is the DNS zone associated to the home network. This zone is set by the CPE and essentially contains the bindings between names and IP addresses of the nodes of the home network. In this document, the CPE does neither perform any DNSSEC management operations such as zone signing nor provide an authoritative service for the zone. Both are delegated to the Public Authoritative Server. The CPE synchronizes the DNS Homenet Zone with the Public Authoritative Server via a hidden master / slave architecture. The Public Authoritative Server might use specific servers for the synchronization of the DNS Homenet Zone: the Public Authoritative Name Server Set.

- DNS Homenet Zone Template: The template used as a basis to generate the DNS Homenet Zone.

- DNS Template Server: The DNS server that hosts the DNS Homenet Zone Template.

- DNS Homenet Reverse Zone: The reverse zone file associated to the DNS Homenet Zone.

- Public Authoritative Master(s): are the visible name server hosting the DNS Homenet Zone. End users' resolutions for the Homenet Domain are sent to this server, and this server is a master for the zone.

- Public Authoritative Name Server Set: is the server the CPE synchronizes the DNS Homenet Zone. It is configured as a slave and the CPE acts as master. The CPE sends information so the DNSSEC zone can be set and served.

- Reverse Public Authoritative Master(s): are the visible name server hosting the DNS Homenet Reverse Zone. End users' resolutions for the Homenet Domain are sent to this server, and this server is a master for the zone.

- Reverse Public Authoritative Name Server Set: is the server the CPE synchronizes the DNS Homenet Reverse Zone. It is configured as a slave and the CPE acts as master. The CPE sends information so the DNSSEC zone can be set and served.

3. Introduction

CPEs are usually constraint 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. One way to avoid exposing CPE is to outsource the authoritative service to a third party. 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 which may be unappropriated for most end users. To leverage this issue, this document proposes DHCP Options so any agnostic CPE can automatically proceed to the appropriated configuration and outsource the authoritative naming service for the home network. This document shows that in most cases, these DHCP Options make outsourcing to a third party (be it the ISP or any ISP independent service provider) transparent for the end user.

When the CPE is plugged, the DHCP Options described in the document enable the CPE:

- 1. To build the DNS Homenet Zone: Building the DNS Homenet Zone requires filling the zone with appropriated bindings likes name / IP addresses of the different devices in the home networks. Such information can be provided for example by the DHCP Server hosted on the CPE. On the other hand, it also requires configuration parameters like the name of the Registered Domain Name associated to the home network or the Public Authoritative Master(s) the DNS Homenet Zone is outsourced to. These configuration parameters are stored in the DNS Homenet Zone Template. This document describes the DHCP Zone Template Option. This option carries a DNS Homenet Zone Template FQDN. In order to retrieve the DNS Homenet Zone Template, the CPE sends a query of type AXFR [RFC1034] [RFC5936]for the DNS Homenet Zone Template FQDN.

- 2. To upload the DNS(SEC) Homenet Zone to the appropriated server: This server is designated as the Public Authoritative Name Server Set. It is in charge of publishing the DNS(SEC) Homenet Zone on the Public Authoritative Master(s). This document describes the DHCP Public Authoritative Name Server Set Option that provides the FQDN of the appropriated server. Note that, in the document we do not consider whether the DNS(SEC) Homenet Zone is signed or not and if signed who signs it. Such questions are out of the scope of the current document.

- 3. To upload the DNS Homenet Reverse Zone to the appropriated server: This server is designated as the Reverse Public Authoritative Name Server Set. It is in charge of publishing the DNS Homenet Reverse Zone on the Reverse Public Authoritative Master(s). This document describes the DHCP Reverse Public Authoritative Name Server Set Option that provides the FQDN of the appropriated server. Similarly to item 2., we do not consider in this document if the DNS 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 DNS Homenet Zone Template, the DNS(SEC) Homenet Zone and DNS 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 DHCP Public Key Option. This option is sent by the CPE to the DHCP Server and provides the Public Key the CPE uses to authenticate itself. The DHCP Server is then responsible to provide the Public Key to the various DNS servers.

As a result, the DHCP Options described in this document enable an agnostic CPE to outsource its naming infrastructure without any configuration from the end user. The main reason no configuration is required by the end user is that there are privileged links: first between the CPE and the DHCP Server and then between the DHCP Server and the various DNS servers (DNS Homenet Zone Server, the Reverse Public Authoritative Name Server Set, Public Authoritative Name Server Set). This enables the CPE to send its authentication credentials (a Public Key) to the DHCP Server that in turn forward it to the various DNS servers. With the authentication credential on the DNS servers set, the CPE is able to update the various zones in a secure way.

If the DHCP Server cannot provide the public key to one of these servers (most likely the Public Authoritative Name Server Set) and the CPE needs to interact with the server, then, the end user is expected to provide the CPE's public key to these servers (the Reverse Public Authoritative Name Server Set or the Public Authoritative Name Server Set) 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 as follows. Section 4 provides an overview of the DHCP 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 DHCP Options and Section 7 details how DHCP Client DHCP Server and DHCP Relay 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 how the CPE is expect to interact with various entities, as well as how the CPE is expected to be configured via DHCP Options. Section 4.1 describes the entities the CPE is expected to interact with. Interaction with each entities is defined via DHCP Options that are expected to configure the CPE. Once configured, the CPE is expected to be able to update some DNS Data hosted by the different entities. As a result security and updating mechanisms play an important role in the specification. Section 4.2 provides an overview of the different security mechanisms considered for securing the CPE transactions and Section 4.3

considers the different update mechanisms considered for the CPE to update the DNS Data.

4.1. Architecture and DHCP Options Overview

This section illustrates how a CPE configures its naming infrastructure to outsource its authoritative naming service. All configurations and settings are performed using DHCP Options. This section, for the sake of simplicity, assumes that the DHCP Server is able to communicate to the various DNS servers and to provide them the public key associated to the CPE. Once each server got the public key, the CPE can proceed to updates in a 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 Public Authoritative Name Server Set 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 Public Key to the DHCP Server using a DHCP Public Key Option (OPTION_PUBLIC_KEY) and sends a DHCP Option Request Option (ORO) for the DHCP Zone Template Option (OPTION_DNS_ZONE_TEMPLATE), the DHCP Public Authoritative Name Server Set Option (OPTION_NAME_SERVER_SET) and the DHCP Reverse Public Authoritative Name Server Set Option (OPTION_REVERSE_NAME_SERVER_SET).

- 2: The DHCP Server makes the Public Key available to the DNS servers, so the CPE can secure its DNS transactions. Note that the 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 were the DHCP Server forwards the set (Public Key, Zone Template FQDN) to the DNS Template Server, the set (Public Key, IPv6 subnet) to the Reverse Public Authoritative Name Server Set and the set (Public Key, Registered Homenet Domain) to the Public Authoritative Name Server Set.

- 3.: The DHCP Server responds to the CPE with the requested DHCP Options, i.e. the DHCP Public Key Option (OPTION_PUBLIC_KEY), DHCP Zone Template Option OPTION_DNS_ZONE_TEMPLATE, DHCP Public

Authoritative Name Server Set Option (OPTION_NAME_SERVER_SET), DHCP Reverse Public Authoritative Name Server Set Option (OPTION_REVERSE_NAME_SERVER_SET).

- 4.: Upon receiving the DHCP Zone Template Option (OPTION_DNS_ZONE_TEMPLATE), the CPE performs an AXFR DNS query for the Zone Template FQDN. The exchange is secured according to the security protocols defined in the Security field of the DHCP option. Once the CPE has retrieved the DNS Zone Template, the CPE can build the DNS Homenet Zone and the DNS Homenet Reverse Zone. Eventually the CPE signs these zones.

- 5.: Once the DNS(SEC) Homenet Reverse Zone has been set, the CPE uploads the zone to the Reverse Public Authoritative Name Server Set. The DHCP Reverse Public Authoritative Name Server Set Option (OPTION_REVERSE_NAME_SERVER_SET) provides the Reverse Public Authoritative Name Server Set FQDN as well as the upload method, and the security protocol to secure the upload.

- 6.: Once the DNS(SEC) Homenet Zone has been set, the CPE uploads the zone to the Public Authoritative Name Server Set. The DHCP Public Authoritative Name Server Set Option (OPTION_NAME_SERVER_SET) provides the Public Authoritative Name Server Set FQDN as well as the upload method and the security protocol to secure the upload.

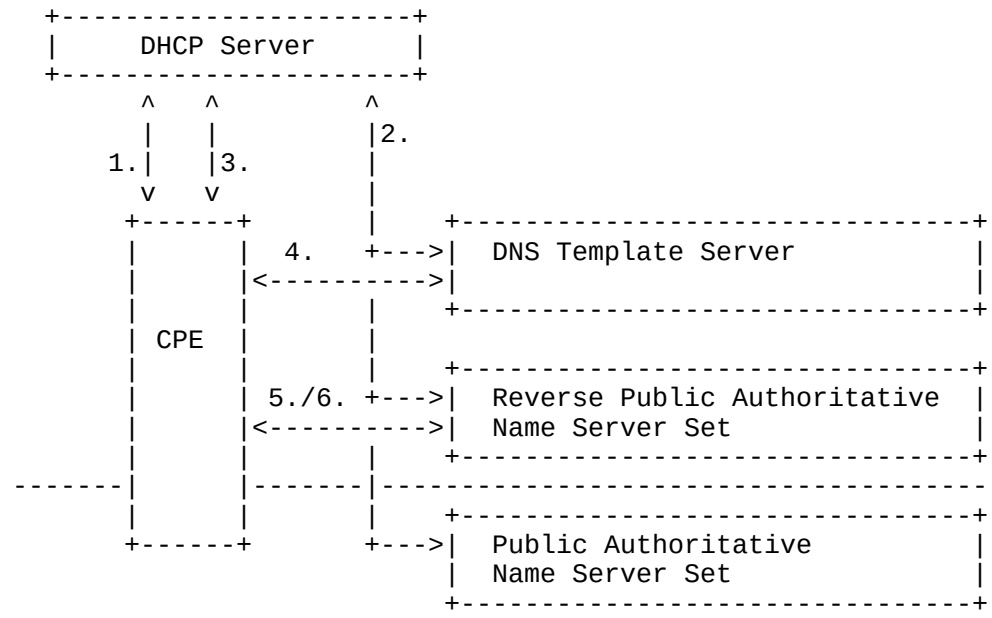


Figure 1: Protocol Overview

As described above, the CPE is likely to interact with various DNS content. This section is focused on DNS Data the CPE is likely to update. More specifically, the CPE is likely to update the:

- DNS Homenet Zone Template: may be updated by the CPE if the configuration of the zone may be changed. This can include additional Public Authoritative Master(s), a different Registered Homenet Domain as the one initially proposed, or a redirection to another domain.

- DNS Homenet Reverse Zone: may be updated every time a new device is connected or dis-connected.

- DNS Homenet Zone: may be updated every time a new device is connected, dis-connected.

In fact, the CPE must be able to perform these updates in a secure manner. There are multiple ways to secure a DNS transaction and this document considers two mechanisms to update a DNS Data (nsupdate and master/slave synchronization). Which security mechanism to use to secure a DNS transaction depends on the expected security (authentication of the authoritative server, mutual authentication, confidentiality...). The expected security may also depends on the kind of transaction performed by the CPE. Section 4.2 describes the

different security mechanisms considered in the document as well as their respective goals. Which mechanism to use to update the DNS Data depends on the kind of update. Frequency of the update, size of the DNS Data to update may be some useful criteria. Section 4.3 positions the nsupdate and master/slave synchronization mechanisms.

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 restricts the scope of security protocols to those 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 provides means to distribute shared secret for example using a specific DHCP 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).

Exchange with the DNS Template Server to retrieve the DNS 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 prevents someone else to query the DNS 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 DNS Homenet Zone. In fact, these DNSSEC related fields are associated to the DNS Homenet Zone Template and not the DNS Homenet Zone.

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

4.3. Master / Slave 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 master / slave synchronization.

The DNS Homenet Zone Template can only be updated with DNS update. The reason is that the DNS Homenet Zone Template contains static configuration data that is not expected to evolve over time.

The DNS Homenet Reverse Zone and the DNS Homenet Zone can be updated either with DNS update or using a master / slave synchronization. As these zones may be large, with frequent updates, we recommend to use the master / slave architecture as described in [I-D.ietf-homenet-front-end-naming-delegation]. The master / slave mechanism is preferred as it better scales and avoids DoS attacks: First the master notifies the slave the zone must be updated, and leaves the slave to proceed to the update when possible. Then, the NOTIFY message sent by the master is a small packet that is less likely to load the slave. At last, the AXFR query performed by the slave is a small packet sent over TCP (section 4.2 [RFC5936]) which makes unlikely the slave to perform reflection attacks with a forged NOTIFY. On the other hand, DNS updates can use UDP, packets require more processing then a NOTIFY, and they do not provide the server the opportunity to post-pone the update.

5. CPE Configuration

5.1. CPE Master / Slave Synchronization Configurations

The master / slave architecture is described in [I-D.ietf-homenet-front-end-naming-delegation]. The CPE is configured as a master whereas the DNS Server is configured as a slave. The DNS Server represents the Public Authoritative Name Server Set or the Reverse Public Authoritative Name Server Set.

When the CPE is plugged its IP address may be unknown to the slave. The section details how the CPE or master communicate the necessary information to set up the slave.

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

5.1.1. CPE / Public Authoritative Name Server Set

The CPE knows the zone to be synchronized by reading the Registered Homenet Domain in the DNS Homenet Zone Template provided by the DHCP Zone Template Option (OPTION_DNS_ZONE_TEMPLATE). The IP address of the slave is provided by the DHCP Public Authoritative Name Server Set Option (OPTION_NAME_SERVER_SET).

The Public Authoritative Name Server Set has been configured with the Registered Homenet Domain and the Public Key that identifies the CPE. The only thing missing 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 DNS Homenet Zone, it sends a NOTIFY message to the Public Authoritative Name Server Sets. Upon receiving the NOTIFY message, the slave reads the Registered Homenet Domain and checks the NOTIFY is sent by the authorized master. This can be done using the shared secret (TSIG) or the public key (SIG(0)). Once the NOTIFY has been authenticated, the Public Authoritative Name Server Sets might consider the source IP address of the NOTIFY query to configure the masters attributes.

5.1.2. CPE / Reverse Public Authoritative Name Server Set

The CPE knows the zone to be synchronized by looking at its assigned prefix. The IP address of the slave is provided by the DHCP Reverse Public Authoritative Name Server Set Option (OPTION_REVERSE_NAME_SERVER_SET).

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

5.2. CPE DNS Data Handling and Update Policies

5.2.1. DNS Homenet Zone Template

The DNS Homenet Zone Template contains at least the related fields of the Public Authoritative Master(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 Master(s). This default settings should provide the CPE the necessary pieces of information to set the homenet naming architecture.

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

The DNS Homenet Zone Template might 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 DNS Homenet Zone Template. Policies that defines what is accepted or rejected is out of scope of this document. However, in this document we assume the Registered Homenet Domain is used as an index by the Public Authoritative Name Server Set, and SIG(0), TSIG are used to authenticate the CPE. As a result, the Registered Homenet Domain should not be modified unless the Public Authoritative Name Server Set can handle with it.

5.2.2. DNS (Reverse) Homenet Zone

The DNS Homenet Zone might be generated from the DNS Homenet Zone Template. How the DNS Homenet Zone is generated is out of scope of this document. In some cases, the DNS Homenet Zone might be the exact copy of the DNS Homenet Zone Template. In other cases, it might be generated from the DNS Homenet Zone Template with additional RRsets. In some other cases, the DNS Homenet Zone might be generated without considering the DNS 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 DNS Homenet Zone Template. This constrain 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 DNS 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 DHCP Options. A few DHCP Options are used to advertise a server the CPE may be expect to interact with. Interaction may require to define how the update is expected to be performed as well as how the communication is secured. Security and Update are shared by multiple DHCP Options and are described in separate sections. Section 6.1 describes the security field, Section 6.2 describes the update fields, the remaining sections Section 6.3, Section 6.4, Section 6.5, Section 6.6 describe the DHCP Options.

6.1. Security Field

The Security Field of the DHCP Option is represented in Figure 2. It indicates the security mechanism 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.

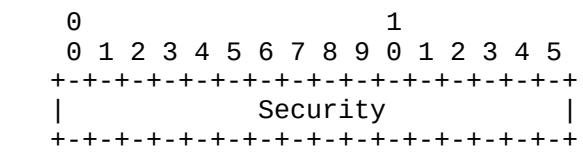


Figure 2: Security Field

- 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 DNS 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 ignored by the DHCP Client.

A Security field with all bits set to zero indicates the operation is not permitted. The Security 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 DHCP Option is represented in Figure 3. It indicates the update mechanism supported by the DNS server. See Section 4.3 for more details.

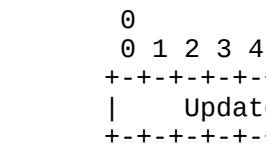


Figure 3: Update Field

- Master / Slave (Bit 0): indicates, when set to 1, that DNS Server supports data synchronization using a Master / Slave 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 DHCP Server and ignored by the DHCP Client.

6.3. DHCP Public Key Option

The DHCP Public Key Option (OPTION_PUBLIC_KEY) indicates the Public Key that is used to authenticate the CPE.

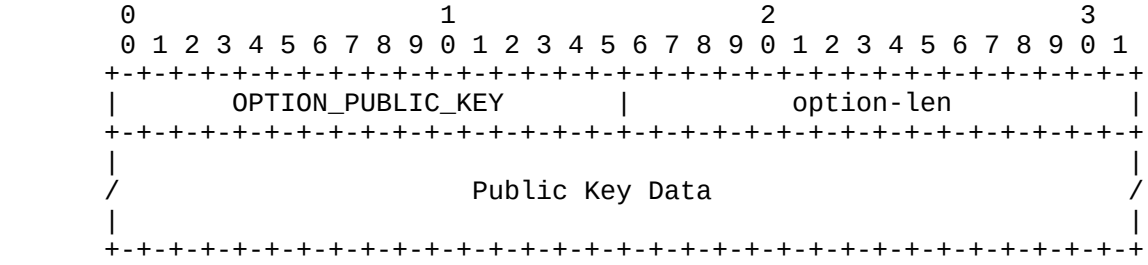


Figure 4: DHCP Public Key Option

- OPTION_PUBLIC_KEY (variable): the option code for the DHCP Public Key Option.

- option-len (16 bits): length in octets of the option-data field as described in [RFC3315].

- Public Key Data: contains the Public Key. The format is the DNSKEY RDATA format as defined in [RFC4034].

6.4. DHCP Zone Template Option

The DHCP Zone Template Option (OPTION_DNS_ZONE_TEMPLATE) Option indicates the CPE how to retrieve the DNS Homenet Zone Template. It provides a FQDN the CPE SHOULD query with a DNS query of type AXFR.

The option also specifies which security protocols are available on the authoritative server. DNS Homenet Zone Template update, if permitted MUST use the DNS Update mechanism.

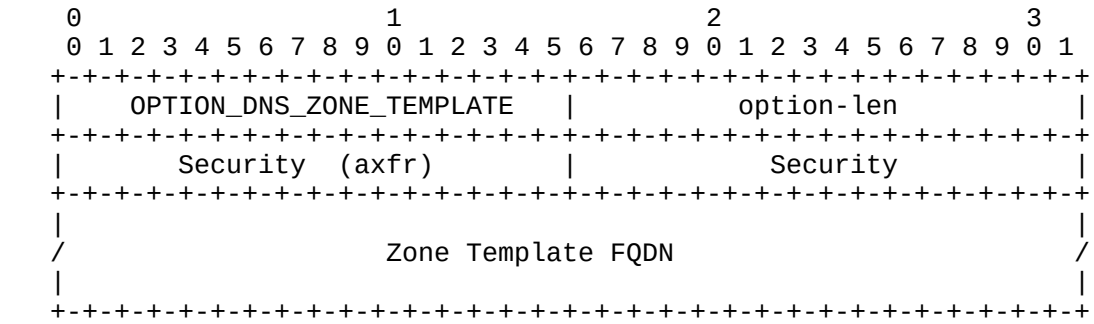


Figure 5: DHCP Zone Template Option

- OPTION_DNS_ZONE_TEMPLATE (variable): the option code for the DHCP Zone Template Option.

- option-len (16 bits): length in octets of the option-data field as described in [RFC3315].

- Security (axfr) (16 bits): defines which security protocols 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.

- Security (16 bits): defines which security protocols 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 DNS Homenet Zone Template.

6.5. DHCP Public Authoritative Name Server Set Option

The DHCP Public Authoritative Name Server Set Option (OPTION_NAME_SERVER_SET) provides information so the CPE can upload the DNS Homenet Zone to the Public Authoritative Name Server Set. Finally, the option provides the security mechanisms that are available to perform the upload. The upload is performed via a DNS master / slave architecture or DNS updates.

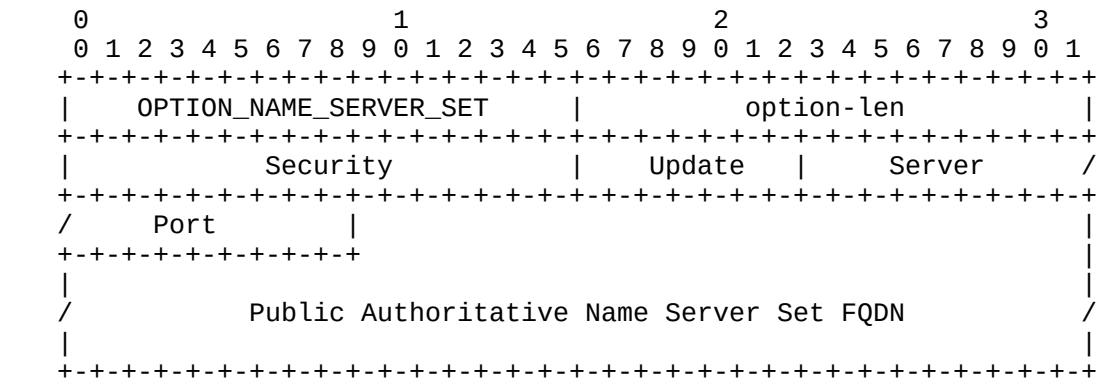


Figure 6: DHCP Public Authoritative Name Server Set Option

- OPTION_NAME_SERVER_SET (16 bits): the option code for the DHCP Public Authoritative Name Server Set Option.

- option-len (16 bits): length in octets of the option-data field as described in [RFC3315].

- Security (16 bits): defines which security protocols 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 Public Authoritative Name Server Set is listening.

- Public Authoritative Name Server Set FQDN (variable): the FQDN of the Public Authoritative Name Server Set.

6.6. DHCP Reverse Public Authoritative Name Server Set Option

The DHCP Reverse Public Authoritative Name Server Set Option (OPTION_REVERSE_NAME_SERVER_SET) provides information so the CPE can upload the DNS Homenet Zone to the Public Authoritative Name Server Set. The option provides the security mechanisms that are available to perform the upload. The upload is performed via a DNS master / slave architecture or DNS updates.

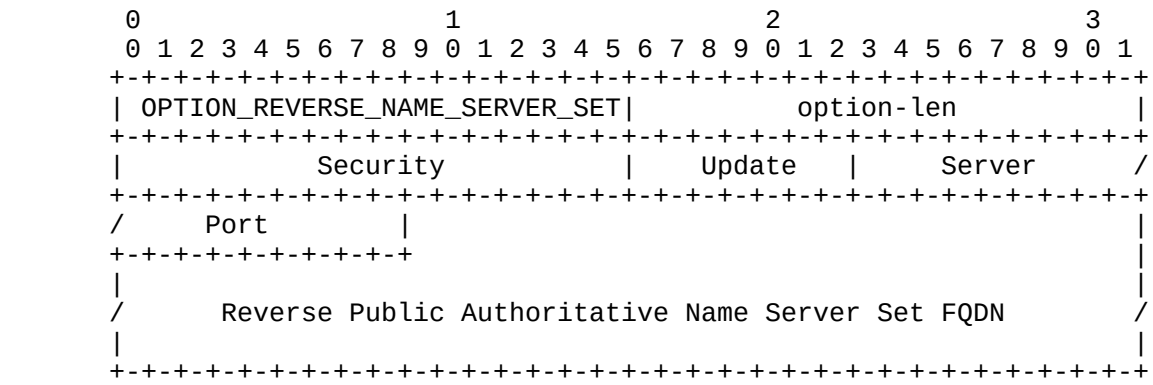


Figure 7: DHCP Reverse Public Authoritative Name Server Set Option

- OPTION_REVERSE_NAME_SERVER_SET (16 bits): the option code for the DHCP Reverse Public Authoritative Name Server Set Option.

- option-len (16 bits): length in octets of the option-data field as described in [RFC3315].

- Security (16 bits): defines which security protocols 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 Public Authoritative Name Server Set is listening.

- Reverse Public Authoritative Name Server Set FQDN (variable): The FQDN of the Reverse Public Authoritative Name Server Set.

7. DHCP Behavior

7.1. DHCPv6 Server Behavior

The DHCP Server sends the DHCP Zone Template Option (OPTION_DNS_ZONE_TEMPLATE), DHCP Public Authoritative Name Server Set Option (OPTION_NAME_SERVER_SET), DHCP Reverse Public Authoritative Name Server Set Option (OPTION_REVERSE_NAME_SERVER_SET) upon request by the DHCP Client.

The DHCP Server MAY receive a DHCP Public Key Option (OPTION_PUBLIC_KEY) from the CPE. Upon receipt of this DHCP Option, the DHCP Sever is expect to communicate this credential to the available DNS Servers like the DNS Template Server, the Public

Authoritative Name Server Set and the Reverse Public Authoritative Name Server Set.

7.2. DHCPv6 Client Behavior

The DHCP Client MAY send a DHCP Public Key Option (OPTION_PUBLIC_KEY) to the DHCP Server. This Public Key authenticates the CPE.

The DHCP Client sends a DHCP Option Request Option (ORO) with the necessary DHCP options.

A CPE SHOULD only send the an ORO request for DHCP Options it needs or for information that needs to be up-to-date.

Upon receiving a DHCP option described in this document, the CPE SHOULD retrieve or update DNS zones using the associated security and update protocols.

7.3. DHCPv6 Relay Behavior

DHCP Relay behavior are not modified by this document.

8. IANA Considerations

The DHCP options detailed in this document is:

- OPTION_DNS_ZONE_TEMPLATE: TBD

- OPTION_NAME_SERVER_SET: TBD

- OPTION_REVERSE_NAME_SERVER_SET: TBD

- OPTION_PUBLIC_KEY: TBD

9. Security Considerations

9.1. DNSSEC is recommended to authenticate DNS hosted data

It is recommended that the (Reverse) DNS 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 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.

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

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. Acknowledgment

We would like to thank Tomasz Mrugalski, Marcin Siodelski and Bernie Volz for their comments on the design of the DHCP 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.

11. References

11.1. Normative References

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

[RFC1996] Vixie, P., "A Mechanism for Prompt Notification of Zone Changes (DNS NOTIFY)", RFC 1996, August 1996.

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

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

[RFC2181] Elz, R. and R. Bush, "Clarifications to the DNS Specification", RFC 2181, July 1997.

[RFC2845] Vixie, P., Gudmundsson, O., Eastlake, D., and B. Wellington, "Secret Key Transaction Authentication for DNS (TSIG)", RFC 2845, May 2000.

[RFC2930] Eastlake, D., "Secret Key Establishment for DNS (TKEY RR)", RFC 2930, September 2000.

[RFC2931] Eastlake, D., "DNS Request and Transaction Signatures (SIG(0)s)", RFC 2931, September 2000.

[RFC3007] Wellington, B., "Secure Domain Name System (DNS) Dynamic Update", RFC 3007, 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.

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, March 2005.

[RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, March 2005.

[RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, March 2005.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC5936] Lewis, E. and A. Hoenes, "DNS Zone Transfer Protocol (AXFR)", RFC 5936, June 2010.

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

[RFC6672] Rose, S. and W. Wijngaards, "DNAME Redirection in the DNS", RFC 6672, June 2012.

[RFC7296] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., and T. Kivinen, "Internet Key Exchange Protocol Version 2 (IKEv2)", STD 79, RFC 7296, October 2014.

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-homenet-front-end-naming-delegation]
Migault, D., Cloetens, W., Griffiths, C., and R. Weber, "Outsourcing Home Network Authoritative Naming Service", draft-ietf-homenet-front-end-naming-delegation-00 (work in progress), September 2014.

[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 Master(s) the ISP is able to build the DNS Homenet Zone Template.

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

When the CPE is plugged (at least the first time), it provides its 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:

- Public Authoritative Name Server Set

- Reverse Public Authoritative Name Server Set

- DNS Template Server

The CPE can update the zone using DNS update or a master / slave 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 Public Authoritative Name Server Set and Public Authoritative Master(s) to a third party. All other DNS Servers DNS Template Server, Reverse Public Authoritative Master(s) and Reverse Public Authoritative Name Server Set remain managed by the ISP. The reason we consider that Reverse Public Authoritative Masters(s) and Reverse Public Authoritative Name Server Set 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 Public Authoritative Name Server Set and Public Authoritative Master(s) to a third party. Similarly, the DNS Template Server can be easily outsourced as detailed in this section

Outsourcing Public Authoritative Name Server Set and Public Authoritative Master(s) requires:

- 1) Updating the DNS 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 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 Master(s), with a button "Configure DNS Homenet Zone Template".

- 2) Updating the DHCP Server Information. In fact the Reverse Public Authoritative Name Server Set 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 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 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 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 DHCP Public Key Option as well as a request for a DHCP Zone Template Option, a DHCP Public Authoritative Name Server Set Option and a DHCP Reverse Public Authoritative Name Server Set 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 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 DHCP 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 DHCP 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 Public Authoritative Name Server Set and Public Masters 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 DHCP 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]

-04: Working Version Major modifications are:

- Re-structuring the draft: description and comparison of update and security mechanisms have been intergrated 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 DHCP 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 DHCP 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 DHCP 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: mglt.ietf@gmail.com

Wouter Cloetens
SoftAtHome
vaartdijk 3 701
3018 Wijgmaal
Belgium

Email: wouter.cloetens@softathome.com

Chris Griffiths
Dyn
150 Dow Street
Manchester, NH 03101
US

Email: cgriffiths@dyn.com
URI: http://dyn.com

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

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

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 12, 2015

P. Pfister
B. Paterson
Cisco Systems
J. Arkko
Ericsson

February 8, 2015

Distributed Prefix Assignment Algorithm
draft-ietf-homenet-prefix-assignment-03

Abstract

This document specifies a distributed algorithm for automatic prefix assignment. Given a set of delegated prefixes, it ensures that at most one prefix is assigned from each delegated prefix to each link. Nodes may assign available prefixes to the links they are directly connected to, or for other private purposes. The algorithm eventually converges and ensures that all assigned prefixes do not overlap.

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 August 12, 2015.

Copyright Notice

Table of Contents

1. Introduction

This document specifies a distributed algorithm for automatic prefix assignment. Given a set of delegated prefixes, nodes may assign available prefixes to links they are directly connected to, or for their private use. The algorithm ensures that the following assertions are eventually true:

1. At most one prefix from each delegated prefix is assigned to each link.

2. Assigned prefixes are not included in and do not include other assigned prefixes.

3. Assigned prefixes do not change in the absence of topology or configuration changes.

In the rest of this document the two first conditions are referred to as the correctness conditions of the algorithm while the third condition is referred to as its convergence condition.

Each assignment has a priority specified by the node making the assignment, allowing for more advanced assignment policies. When multiple nodes assign different prefixes from the same delegated prefix to the same link, or when multiple nodes assign overlapping prefixes, the assignment with the highest priority is kept and other assignments are removed.

The prefix assignment algorithm requires that participating nodes share information through a flooding mechanism. If the flooding mechanism ensures that all messages are propagated to all nodes faster than a given timing upper bound, the algorithm also ensures that all assigned prefixes used for networking operations (e.g., host configuration) remain unchanged, unless another node assigns an overlapping prefix with a higher assignment priority, or the topology changes and renumbering cannot be avoided.

2. Terminology

In this document, the key words "MAY", "MUST, "MUST NOT", "OPTIONAL", and "SHOULD", are to be interpreted as described in [RFC2119].

This document makes use of the following terminology:

Link: An object the distributed algorithm will assign prefixes to. A Node may only assign prefixes to Links it is directly connected to. A Link is either Shared or Private.

Private Link: A Private Link is an abstract concept defined for the sake of this document. It allows nodes to make assignments for their private use or delegation. For instance, every DHCPv6-PD [RFC3633] client MAY be considered as a different Private Link.

Shared Link: A Link multiple nodes may be connected to. Most of the time, a Shared Link would consist in a multi-access link or point-to-point link, virtual or physical, requiring prefixes to be assigned to.

Delegated Prefix: A prefix provided to the algorithm and used as a prefix pool for Assigned Prefixes.

Node ID: A value identifying a given participating node. The set of identifiers MUST be strictly and totally ordered (e.g., using the alphanumeric order).

Flooding Mechanism: A mechanism implementing reliable broadcast and used to advertise published Assigned Prefixes.

Flooding Delay: Value which SHOULD be provided by the Flooding Mechanism indicating a deterministic or likely upper bound of the information propagation delay. When the Flooding Mechanism does not provide a value, it is set to DEFAULT_FLOODING_DELAY (Section 7).

Advertised Prefix: A prefix advertised by another node and delivered to the local node by the Flooding Mechanism. It has an Advertised Prefix Priority and, when assigned to a directly connected Shared Link, is associated with a Shared Link.

Advertised Prefix Priority: A value that defines the priority of an Advertised Prefix received from the Flooding Mechanism or a published Assigned Prefix. Whenever multiple Advertised Prefixes are conflicting, all Advertised Prefixes but the one with the greatest priority will eventually be removed. In case of tie, the assignment advertised by the node with the greatest Node ID is kept and others are removed. In order to ensure convergence, the range of priority values MUST have an upper bound.

Assigned Prefix: A prefix included in a Delegated Prefix and assigned to a Shared or Private Link. It represents a local decision to assign a given prefix from a given Delegated Prefix to a given Link. The algorithm ensures that there never is more than one Assigned Prefix per Delegated Prefix and Link pair. When destroyed, an Assigned Prefix is set as not applied, ceases to be advertised, and is removed from the set of Assigned Prefixes.

Applied (Assigned Prefix): When an Assigned Prefix is applied, it MAY be used (e.g., for host configuration, routing protocol configuration, prefix delegation). When not applied, it MUST NOT be used for any other purposes than the prefix assignment algorithm. Each Assigned Prefix is associated with a timer (Apply Timer) used to apply the Assigned Prefix. An Assigned Prefix is unapplied when destroyed.

Published (Assigned Prefix): The Assigned Prefix is advertised through the Flooding Mechanism as assigned to its associated Link. A published Assigned Prefix MUST have an Advertised Prefix Priority. It will appear as an Advertised Prefix to other nodes, once received through the Flooding Mechanism.

Prefix Adoption: When an Advertised Prefix which does not conflict with any other Advertised Prefix or published Assigned Prefix stops being advertised, any other node connected to the same Link MAY, after some random delay, start advertising the same prefix. This procedure is called adoption and provides seamless assignment transfer from a node to another, e.g., in case of node failure.

Backoff Timer: Every Delegated Prefix and Link pair is associated with a timer counting down to zero. It is used to avoid multiple nodes from making colliding assignments by delaying the creation of new Assigned Prefixes or the advertisement of adopted Assigned Prefixes by a random amount of time.

Renumbering: Event occurring when an Assigned Prefix which was applied is destroyed. It is undesirable as it usually implies reconfiguring routers or hosts.

3. Applicability statement

Each node MUST have a set of disjoint Delegated Prefixes. This set MAY change over time and be different from one node to another at some point, but nodes MUST eventually have the same set of disjoint Delegated Prefixes.

Given this set of disjoint Delegated Prefixes, nodes may assign available prefixes from each Delegated Prefix to the Links they are directly connected to. The algorithm ensures that at most one prefix from a given Delegated Prefix is assigned to a given Link.

The algorithm can be applied to any address space and can be used to manage multiple address spaces simultaneously. For instance, an implementation can make use of IPv4-mapped IPv6 addresses [RFC4291] in order to manage both IPv4 and IPv6 prefix assignment using a single prefix space.

The algorithm supports dynamically changing topologies:

- Nodes may join or leave the set of participating nodes.
- Nodes may join or leave Links.
- Links may be joined or split.

All nodes MUST run a common Flooding Mechanism in order to share published Assigned Prefixes. The set of participating nodes is defined as the set of nodes participating in the Flooding Mechanism.

The Flooding Mechanism MUST:

- Provide a way to flood Assigned Prefixes assigned to a directly connected Link along with their respective Advertised Prefix Priority and the Node ID of the node which advertises it.
- Specify whether an Advertised Prefix was assigned to a directly connected Shared Link, and if so, on which one.

In addition, a Flooding Delay SHOULD be specified and respected in order to avoid renumbering. If not specified, or whenever the Flooding Mechanism is unable to respect the provided delay, renumbering may happen. As such delay often depends on the size of the network, it MAY change over time and MAY be different from one node to another.

The algorithm ensures that whenever the Flooding Delay is provided and respected, and in the absence of topology change or delegated prefix removal, renumbering never happens.

Each node MUST have a Node ID. Node IDs MAY change over time and be the same on multiple nodes at some point, but each node MUST eventually have a Node ID which is unique among the set of participating nodes.

4. Algorithm Specification

This section specifies the behavior of nodes implementing the prefix assignment algorithm.

4.1. Algorithm Terminology

The algorithm makes use of the following terms:

Current Assignment: For a given Delegated Prefix and Link, the Current Assignment is the Assigned Prefix (if any) included in the Delegated Prefix and assigned to the given Link.

Precedence: An Advertised Prefix takes precedence over an Assigned Prefix if and only if:

* The Assigned Prefix is not published.

* The Assigned Prefix is published and the Advertised Prefix Priority from the Advertised Prefix is strictly greater than the Advertised Prefix Priority from the Assigned Prefix.

* The Assigned Prefix is published, the priorities are identical, and the Node ID from the node advertising the Advertised Prefix is strictly greater than the local Node ID.

Best Assignment: For a given Delegated Prefix and Link, the Best Assignment is (if any) the Advertised Prefix:

* Including or included in the Delegated Prefix.

* Assigned on the given Link.

* Having the greatest Advertised Prefix Priority among Advertised Prefixes assigned on the given Link (and, in case of tie, the prefix advertised by the node with the greatest Node ID among all prefixes with greatest priority).

* Taking precedence over the Current Assignment associated with the same Link and Delegated Prefix (if any).

Valid (Assigned Prefix) An Assigned Prefix is valid if and only if the two following conditions are met:

* No Advertised Prefix including or included in the Assigned Prefix takes precedence over the Assigned Prefix.

* No Advertised Prefix including or included in the same Delegated Prefix as the Assigned Prefix and assigned to the same Link takes precedence over the Assigned Prefix.

4.2. Prefix Assignment Algorithm Routine

This section specifies the prefix assignment algorithm routine. It is defined for a given Delegated Prefix/Link pair and may be run either as triggered by the Backoff Timer, or not.

For a given Delegated Prefix and Link pair, the routine MUST be run as not triggered by the Backoff Timer whenever:

- An Advertised Prefix including or included in the considered Delegated Prefix is added or removed.

- An Assigned Prefix included in the considered Delegated Prefix and associated with a different Link than the considered Link was destroyed, while there is no Current Assignment associated with the given pair. This case MAY be ignored if the creation of a new Assigned Prefix associated with the considered pair is not desired.

- The considered Delegated Prefix is added.

- The considered Link is added.

- The Node ID is modified.

Additionally, for a given Delegated Prefix and Link pair, the routine MUST be run as triggered by the Backoff Timer whenever:

- The Backoff Timer associated with the considered Delegated Prefix/Link pair fires while there is no Current Assignment associated with the given pair.

When such an event occurs, a node MAY delay the execution of the routine instead of executing it immediately, e.g. while receiving an update from the Flooding Mechanism, or for security reasons (see Section 8). Even though other events occur in the meantime, the routine MUST be run only once. It is also assumed that, whenever one of these events is the Backoff Timer firing, the routine is executed as triggered by the Backoff Timer.

In order to execute the routine for a given Delegated Prefix/Link pair, first look for the Best Assignment and Current Assignment associated with the Delegated Prefix/Link pair, then execute the corresponding case:

1. If there is no Best Assignment and no Current Assignment: Decide whether the creation of a new assignment for the given Delegated Prefix/Link pair is desired (As any result would be valid, the way the decision is taken is out of the scope of this document) and do the following:

   * If it is not desired, stop the execution of the routine.

   * Else if the Backoff Timer is running, stop the execution of the routine.

   * Else if the routine was not executed as triggered by the Backoff Timer, set the Backoff Timer to some random delay between ADOPT_MAX_DELAY and BACKOFF_MAX_DELAY (see Section 7) and stop the execution of the routine.

   * Else, continue the execution of the routine.

   Select a prefix for the new assignment (see Section 5 for guidance regarding prefix selection). This prefix MUST be included in or be equal to the considered Delegated Prefix and MUST NOT include or be included in any Advertised Prefix. If a suitable prefix is found, use it to create a new Assigned Prefix:

   * Assigned to the considered Link.

   * Not applied.

   * The Apply Timer set to ’2 * Flooding Delay’.

   * Published with some selected Advertised Prefix Priority.

2. If there is a Best Assignment but no Current Assignment: Cancel the Backoff Timer and use the prefix from the Best Assignment to create a new Assigned Prefix:

   * Assigned to the considered Link.

   * Not applied.

   * The Apply Timer set to '2 * Flooding Delay'.

   * Not published.

3. If there is a Current Assignment but no Best Assignment:

   * If the Current Assignment is not valid, destroy it, and execute the routine again, as not triggered by the Backoff Timer.

   * If the Current Assignment is valid and published, stop the execution of the routine.

   * If the Current Assignment is valid and not published, the node MAY either:

     + Adopt the prefix by cancelling the Apply Timer and set the Backoff Timer to some random delay between 0 and ADOPT_MAX_DELAY (see Section 7). This procedure is used to avoid renumbering when the node advertising the prefix left the Shared Link.

     + Destroy it and execute case 1 in order to create a different assignment.

4. If there is a Current Assignment and a Best Assignment:

   * Cancel the Backoff Timer.

   * If the two prefixes are identical, set the Current Assignment as not published. If the Current Assignment is not applied and the Apply Timer is not set, set the Apply Timer to '2 * Flooding Delay'.

   * If the two prefixes are not identical, destroy the Current Assignment and go to case 2.

When the prefix assignment algorithm routine requires an assignment to be created or adopted, any Advertised Prefix Priority value can be used. Other documents MAY provide restrictions over this value

depending on the context the algorithm is operating in, or leave it as implementation-specific.

When the prefix assignment algorithm routine requires an assignment to be created or adopted, the chosen Advertised Prefix Priority is unspecified (any value would be valid). The values to be used in such situations MAY be specified by other documents making use of the prefix assignment algorithm or be left as an implementation specific choice.

4.3. Overriding and Destroying Existing Assignments

In addition to the behaviors specified in Section 4.2, the following procedures MAY be used in order to provide more advanced behavior (Section 6):

Overriding Existing Assignments: For any given Link and Delegated Prefix, a node MAY create a new Assigned Prefix using a chosen prefix and Advertised Prefix Priority such that:

* The chosen prefix is included in or is equal to the considered Delegated Prefix.

* The Current Assignment, if any, as well as all existing Assigned Prefixes which include or are included inside the chosen prefix, are destroyed.

* It is not applied.

* The Apply Timer set to ’2 * Flooding Delay’.

* It is published.

* The Advertised Prefix Priority is greater than the Advertised Prefix Priority from all Advertised Prefixes which include or are included in the chosen prefix.

In order to ensure algorithm convergence:

* Such overriding assignments MUST NOT be created unless there was a change in the node configuration, a Link was added, or an Advertised Prefix was added or removed.

* The chosen Advertised Prefix Priority for the new Assigned Prefix SHOULD be greater than all priorities from the destroyed Assigned Prefixes. If not, simple topologies with only two nodes may not converge. Nodes which do not respect this rule MUST implement a mechanism which detects whether the

distributed algorithm do not converge and, whenever this would happen, stop creating overriding Assigned Prefixes which do not hold this rule. The specifications for such safety procedures are out of the scope of this document.

Removing an Assigned Prefix: A node MAY destroy any Assigned Prefix which is published. Such an event reflects the desire from a node to not assign a prefix from a given Delegated Prefix to a given Link anymore. In order to ensure algorithm convergence, such procedure MUST NOT be executed unless there was a change in the node configuration. Additionally, whenever an Assigned Prefix is destroyed this way, the prefix assignment algorithm routine MUST be run for the Delegated Prefix/Link pair associated with the deleted Assigned Prefix.

These procedures are OPTIONAL. They could be used for diverse purposes, e.g., for providing custom prefix assignment configuration or reacting to prefix space exhaustion (by overriding short Assigned Prefixes and assigning longer ones).

4.4. Other Events

When the Apply Timer fires, the associated Assigned Prefix MUST be applied.

When the Backoff Timer associated with a given Delegated Prefix/Link pair fires while there is a Current Assignment associated with the same pair, the Current Assignment MUST be published with some associated Advertised Prefix Priority and, if the prefix is not applied, the Apply Timer MUST be set to '2 * Flooding Delay'.

When a Delegated Prefix is removed from the set of Delegated Prefixes, all Assigned Prefixes included in the removed Delegated Prefix MUST be destroyed.

When one Delegated Prefix is replaced by another one that includes or is included in the deleted Delegated Prefix, all Assigned Prefixes which were included in the deleted Delegated Prefix but are not included in the added Delegated Prefix MUST be destroyed. Others MAY be kept.

When a Link is removed, all Assigned Prefixes assigned to that Link MUST be destroyed.

5. Prefix Selection Considerations

When the prefix assignment algorithm routine specified in Section 4.2 requires a new prefix to be selected, the prefix MUST be selected either:

- Among prefixes which were previously assigned and applied on the considered Link. For that purpose, Applied Prefixes may be stored in stable storage along with their associated Link.
- Randomly, picked in a set of at least RANDOM_SET_SIZE (see Section 7) candidate prefixes. If less than RANDOM_SET_SIZE candidates can be found, the prefix MUST be picked among all candidates.
- Based on some custom selection process specified in the configuration.

A simple implementation MAY randomly pick the prefix among all available prefixes, but this strategy is inefficient in terms of address space use as a few long prefixes may exhaust the pool of available short prefixes.

The rest of this section describes a more efficient approach which MAY be applied any time a node needs to pick a prefix for a new assignment. The two following definitions are used:

Available prefix: The prefix A/N is available if and only if A/N does not include and is not included in any Assigned or Advertised Prefix but A/(N-1) does include an Assigned or Advertised Prefix (or N equals 0 and there is no Assigned or Advertised Prefixes at all).

Candidate prefix: A prefix which is included in or is equal to an available prefix.

The procedure described in this section takes the three following criteria into account:

Stability: In some cases, it is desirable that the selected prefix remains the same across executions and reboots. For this purpose, prefixes previously applied on the Link or pseudo-random prefixes generated based on node and Link specific values may be considered.

Randomness: When no stored or pseudo-random prefix is chosen, a prefix may be randomly picked among RANDOM_SET_SIZE candidates of

desired length. If less than RANDOM_SET_SIZE candidates can be found, the prefix is picked among all candidates.

Addressing-space usage efficiency: In the process of assigning prefixes, a small set of badly chosen long prefixes may prevent any shorter prefix from being assigned. For this reason, the set of RANDOM_SET_SIZE candidates is created from the set of available prefixes with longest prefix lengths and, in case of tie, preferring small prefix values.

When executing the procedure, do as follows:

1. For each prefix stored in stable-storage, check if the prefix is included in or equal to an available prefix. If so, pick that prefix and stop.

2. For each prefix length, count the number of available prefixes of the given length.

3. If the desired prefix length was not specified, select one. The available prefixes count computed previously may be used to help picking a prefix length such that:

   * There is at least one candidate prefix.

   * The prefix length is chosen great enough to not exhaust the address space.

   Let N be the chosen prefix length.

4. Iterate over available prefixes starting with prefixes of length N down to length 0 and create a set of RANDOM_SET_SIZE candidate prefixes of length exactly N included in or equal to available prefixes. The end goal here is to create a set of RANDOM_SET_SIZE candidate prefixes of length N included in a set of available prefixes of maximized prefix length. In case of a tie, smaller prefix values (as defined by the bit-wise lexicographical order) are preferred.

5. For each pseudo-random prefix, check if the prefix is equal to a candidate prefix. If so, pick that prefix and stop.

6. Choose a random prefix from the set of selected candidates.

The complexity of this procedure is equivalent to the complexity of iterating over available prefixes. Such operation may be accomplished in linear time, e.g., by storing Advertised and Assigned Prefixes in a binary trie.

6. Implementation Capabilities and Node Behavior

Implementations of the prefix assignment algorithm may vary from very basic to highly customizable, enabling different types of fully interoperable behaviors. The three following behaviors are given as examples:

Listener: The node only acts upon assignments made by other nodes, i.e, it never creates new assignments nor adopt existing ones. Such behavior does not require the implementation of the considerations specified in Section 5 or Section 4.3. The node never checks existing assignments validity, which makes this behavior particularly suited to lightweight devices which can rely on more capable neighbors to make assignments on directly connected Shared Links.

Basic: The node is capable of assigning new prefixes or adopting prefixes which do not conflict with any other existing assignment. Such behavior does not require the implementation of the considerations specified in Section 4.3. It is suited to situations where there is no preference over which prefix should be assigned to which Link, and there is no priority between different Links.

Advanced: The node is capable of assigning new prefixes, adopting existing ones, making overriding assignments and destroying existing ones. Such behavior requires the implementation of the considerations specified in Section 5 and Section 4.3. It is suited when the administrator desires some particular prefix to be assigned on a given Link, or some Links to be assigned prefixes with a greater priority.

7. Algorithm Parameters

This document does not provide values for ADOPT_MAX_DELAY, BACKOFF_MAX_DELAY and RANDOM_SET_SIZE. The algorithm ensures convergence and correctness for any chosen values, even when these are different from node to node. They MAY be adjusted depending on the context, providing a tradeoff between convergence time, efficient addressing, low verbosity (less traffic is generated by the Flooding Mechanism), and low collision probability.

ADOPT_MAX_DELAY (respectively BACKOFF_MAX_DELAY) represents the maximum backoff time a node may wait before adopting an assignment (respectively making a new assignment). BACKOFF_MAX_DELAY MUST be greater than or equal to ADOPT_MAX_DELAY. The greater ADOPT_MAX_DELAY and (BACKOFF_MAX_DELAY - ADOPT_MAX_DELAY), the lower

the collision probability and the verbosity, but the greater the convergence time.

RANDOM_SET_SIZE represents the desired size of the set a random prefix will be picked from. The greater RANDOM_SET_SIZE, the better the convergence time and the lower the collision probability, but the worse the addressing-space usage efficiency.

When the Flooding Mechanism does not provide a Flooding Delay, it is set to DEFAULT_FLOODING_DELAY. As participating nodes do not need to agree on a common Flooding Delay value, this default value MAY be different from one node to another. If the context in which the algorithm is used does not suffer from renumbering, the value 0 MAY be used. Otherwise it depends on the Flooding Mechanism properties and the desired renumbering probability, and is therefore out of scope of this document.

8. Security Considerations

The prefix assignment algorithm functions on top of two distinct mechanisms, the Flooding Mechanism and the Node ID assignment mechanism.

An attacker able to publish Advertised Prefixes through the flooding mechanism may perform the following attacks:

* Publish a single overriding assignment for a whole Delegated Prefix or for the whole address space, thus preventing any node from assigning prefixes to Links.

* Quickly publish and remove Advertised Prefixes, generating traffic at the Flooding Mechanism layer and causing multiple executions of the prefix assignment algorithm in all participating nodes.

* Publish and remove Advertised Prefixes in order to prevent the convergence of the execution.

An attacker able to prevent other nodes from accessing a portion or the whole set of Advertised Prefixes may compromise the correctness of the execution.

An attacker able to cause repetitive Node ID changes may induce traffic generation from the Flooding Mechanism and multiple executions of the prefix assignment algorithm in all participating nodes.

An attacker able to publish Advertised Prefixes using a Node ID used by another node may prevent the correctness and convergence of the execution.

Whenever the security of the Flooding Mechanism and Node ID assignment mechanism could not be ensured, the convergence of the execution may be prevented. In environments where such attacks may be performed, the execution of the prefix assignment algorithm routine SHOULD be rate limited, as specified in Section 4.2.

9. IANA Considerations

This document has no actions for IANA.

10. Acknowledgments

The authors would like to thank those who participated in the previous document's version development as well as the present one. In particular, the authors would like to thank Tim Chown, Fred Baker, Mark Townsley, Lorenzo Colitti, Ole Troan, Ray Bellis, Markus Stenberg, Wassim Haddad, Joel Halpern, Samita Chakrabarti, Michael Richardson, Anders Brandt, Erik Nordmark, Laurent Toutain, Ralph Droms, Acee Lindem and Steven Barth for interesting discussions and document review.

11. References

11.1. Normative References

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

11.2. Informative References

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

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

Appendix A. Static Configuration Example

This section describes an example of how custom configuration of the prefix assignment algorithm may be implemented.

The node configuration is specified as a finite set of rules. A rule is defined as:

- A prefix to be used.
- A Link on which the prefix may be assigned.
- An Assigned Prefix Priority (smallest possible Assigned Prefix Priority if the rule may not override other Assigned Prefixes).
- A rule priority (0 if the rule may not override existing Advertised Prefixes).

In order to ensure the convergence of the execution, the Assigned Prefix Priority MUST be an increasing function (not necessarily strictly) of the configuration rule priority (i.e. the greater is the configuration rule priority, the greater the Assigned Prefix Priority must be).

Each Assigned Prefix is associated with a rule priority. Assigned Prefixes which are created as specified in Section 4.2 are given a rule priority of 0.

Whenever the configuration is changed or the prefix assignment algorithm routine is run: For each Link/Delegated Prefix pair, look for the configuration rule with the highest configuration rule priority such that:

- The prefix specified in the configuration rule is included in the considered Delegated Prefix.
- The Link specified in the configuration rule is the considered Link.
- All the Assigned Prefixes which would need to be destroyed in case a new Assigned Prefix is created from that configuration rule (as specified in Section 4.3) have an associated rule priority which is strictly lower than the one of the considered configuration rule.
- The assignment would be valid when published with an Advertised Prefix Priority equal to the one specified in the configuration rule.

If a rule is found, a new Assigned Prefix is created based on that rule in conformance with Section 4.3. The new Assigned Prefix is associated with the Advertised Prefix Priority and the rule priority specified in the considered configuration rule.

Note that the use of rule priorities ensures the convergence of the execution.

Authors' Addresses

Pierre Pfister
Cisco Systems
Paris
France

Email: pierre.pfister@darou.fr

Benjamin Paterson
Cisco Systems
Paris
France

Email: benjamin@paterson.fr

Jari Arkko
Ericsson
Jorvas 02420
Finland

Email: jari.arkko@piuha.net

Homenet Working Group
Internet-Draft
Intended status: Informational
Expires: September 6, 2015

M. Wasserman
Painless Security
C. Hopps
Deutsche Telekom
J. Chroboczek
University of Paris-Diderot (Paris 7)
March 5, 2015

Homenet IS-IS and Babel Comparison
draft-mrw-homenet-rtg-comparison-02.txt

Abstract

This document is intended to provide information to members of the IETF Home Networks Working Group (Homenet WG), so that we can make an informed decision regarding which routing protocol to use in home networks. The routing protocols compared in this document are: The Babel Routing Protocol (Babel) and The Intermediate System to Intermediate System Intra-Domain Routing Protocol (IS-IS).

Status of This Memo

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

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 6, 2015.

Table of Contents

1. Introduction

This document compares IS-IS (ISO/IEC 10589:2002) [ISO10589] and Babel [RFC6126] according to several criteria related to their use in home networks (Homenets), as defined by the Homenet WG.

[There has been a request that this document compare OSPF as well as Babel and IS-IS. If the WG would find that useful, we would need an OSPF expert who is willing to provide text on OSPF similar to the information that has been provided for IS-IS and Babel.]

Please note that this document does not represent the consenus of any group, not even the authors. It is an organized collection of facts and well-informed opinions provided by experts on Babel and IS-IS that may be useful to the Homenet WG in choosing a routing protocol.

The Homenet environment is different from the environment of a professionally administered network. The most obvious difference is that most home networks are not administered: any protocols used must be robust and fully self-configuring, and any tuning knobs that they provide will be unused in the vast majority of deployments. There may be exceptions to the "no configuration" rule in some environments. For instance, Homenet products may be used in environments where network security is important enough to warrant the configuration of security information, such as keys or certificates. However, even in those environments, minimal configuration should be necessary to deploy a Homenet network.

Another difference is that Homenets are usually built out of a specific class of cheap device, the "Plastic Home Router". A Plastic Home Router’s firmware is installed at the factory, and is most likely never updated. Additionally, experience shows that home routers are often used way beyond their warranty period, and even after their manufacturer leaves the router business. This, again, argues in favour of simple, robust protocols that are easy to implement and can be expected to keep functioning without software updates.

A Plastic Home Router is usually equipped with a reasonable CPU but fairly small amounts of flash and RAM. At the time of writing, a typical example of the bottom of the range has a 200MHz MIPS 4KEc CPU (8kB data cache), but only 4MB of flash and 8MB of RAM. More expensive multi-purpose products may have a 700MHz MIPS 24Kc with 32MB of flash and as much as 128MB of RAM.

We expect smaller devices to want to participate in the Homenet protocols. In particular, some vendors have expressed interest in producing sensor-class hardware that is able to interoperate with

Homenet (smart thermostats, home automation hardware, etc.). It is therefore desirable to be able to scale down the Homenet protocols down to a few tens of kB, at least in a stub role.

Homenets are built and grow organically, and often end up consisting of multiple link technologies with widely different performance characteristics (twisted-pair Ethernet at gigabit speed, IEEE 802.11 (WiFi) and its multiple variants, Powerline Ethernet, etc.). It is desirable for a Homenet routing protocol to be able to dynamically optimise paths according to the link characteristics.

Experts appear to disagree on the expected size of a Homenet: we have heard estimates ranging from just one router up to 250 routers. In any case, while scaling beyond a few thousand nodes is not likely to be relevant to Homenet in the foreseeable future, the Homenet protocols, if successful, will be repurposed to larger networks, whether we like it or not, and it is desirable to use a protocol that scales well.

2. Protocols and Extensions Included in Comparison

Both the IS-IS and Babel protocols are updated and extended over time. This section lists the extensions that were considered in this comparison. Additional protocol extensions could affect some of the information included in this document.

2.1. IS-IS Protocol and Extensions

In addition to the base IS-IS protocol specification [ISO10589], this comparison considers the following IS-IS extensions:

Homenet related specifications

- ISIS Auto-Configuration [ISIS-AUTOCONF];
- Source-Specific routing in IS-IS [ISIS-SS].

Base protocol specifications.

- Use of OSI IS-IS for Routing in TCP/IP and Dual Environments [RFC1195]
- IS-IS Extensions for Traffic Engineering [RFC5305]
- Routing IPv6 with IS-IS [RFC5308]

2.2. Babel Protocol and Extensions

In addition to the base Babel Protocol specification (RFC 6126), this comparison considers the following Babel extensions:

- Extension Mechanism for the Babel Routing Protocol [BABEL-EXT];
- Source-Specific Routing [BABEL-SS], described in more detail in [SS-ROUTING].

3. Routing Algorithms

IS-IS is a Link State routing protocol, and Babel is a Loop-Avoiding Distance Vector routing protocol. There are some differences between these algorithms, particularly in terms of scalability, how much information is exchanged when the routing topology changes, and how far topology changes are propagated.

3.1. Link State Algorithm

Link state algorithms distribute information for each node to all other nodes in the network using a flooding algorithm. This database of information is then used by each node to compute the best path to the other nodes in the network.

One benefit of this algorithm is that each node contains the full knowledge of the topology of the network. This information can be used by other applications outside the routing protocol itself.

Additionally the flooding algorithm has been found as an efficient method for other applications to distribute node-specific application data, although some care must be taken with this use so as not to disrupt the fundamental routing function.

3.2. Loop-Avoiding Distance-Vector Algorithm (Babel)

Distance-vector algorithms distribute information about the path length to reach each destination through a given neighbor. Packets are forwarded through the neighbor that is advertising the best path towards the destination.

Babel, like EIGRP, DSDV, and to a certain extent BGP, uses a loop-avoiding distance-vector algorithm: it avoids creating a loop even during reconvergence, there is no "counting to infinity", and even short-lived "microloops" are avoided in most cases.

3.3. Discussion

From a purely algorithmic point of view, both kinds of algorithms are known to scale well beyond the needs of Homenet. Issues of scaling over wireless and lossy links is discussed in Section 16.

4. Convergence Times

Convergence time is defined as the amount of time after a link failure is detected during which the network is not fully operational. It does not include the time necessary to detect a link failure.

4.1. IS-IS

Given fast flooding of any change in the network, IS-IS has been shown to acheive sub 200ms end-to-end convergence even in very large provider networks (single area 900+ nodes). Basically the time for convergence is the time to propagate new link state from one end of the network to the other plus the SPF (tree computation interval) and FIB loading time. The flooding is done without delay and prior to running the SPF (tree-calculation) algorithm. Thus is roughly proportional to propagation delay across the diameter of the network. The tree calculation is sub 20ms on modern CPUs. FIB load time depends on the FIB hardware and design and not the routing protocol choice.

[According to Gredler, "today, 1000-2000 routers in a single area are said to represent the upper boundary of IS-IS [...] The authors do not endorse these optimistic area numbers, since a lot is dependent on other factors than just the raw number of routers." (p. 84). And I’m pretty sure he’s not thinking of 200MHz CPUs with 8kB data caches, 802.11 and powerline. -- jch]

We easily should expect sub-second convergence for any change in reachability (addition or subtraction) in any conceivable Homenet deployment that does not use IEEE 802.11 for router-to-router links (see Section 16).

4.2. Babel

Since Babel maintains a redundant routing table, it is most often able to reconverge almost instantaneously after a link failure (this is similar to e.g. EIGRP). In the case where no feasible routes are available, Babel reconverges in 20ms per hop to the source.

4.3. Discussion

Both protocols enjoy fast convergence. However, the time needed to reconverge is dwarfed by the amount of time needed to detect a link failure, which is the hold time in IS-IS (30s by default), and 2.5 hello intervals on Babel wired links (2.5*4s, i.e. 10s by default). (Babel performs link quality estimation on wireless links, so the delay is a little higher when wireless links are involved.)

This can be mitigated somewhat by using smaller timers, at the cost of higher overhead, especially on wireless links, which tend to suffer from abominable per-frame overhead. IS-IS supports hold times as small as 1s, while Babel supports hello intervals as low as 10ms (leading to a 25ms hold time). (Either protocol could of course be combined with BFD, which supports arbitrarily small hold times.)

It has been suggested that physical layer carrier sense could be used to detect broken links in a timely manner. Unfortunately, this technique is not useful in Homenet, since most Plastic Home Routers put their Ethernet ports behind an internal switch in order to provide 5 or more Ethernet ports using a single NIC: carrier sense indicates the status of the internal link to the switch, not that of the user-visible Ethernet link. Carrier sense has also been found to be unreliable on wireless links.

5. Autoconfiguration

Most home networks are not administered, so a routing protocol needs to be entirely self-configuring in order to be suitable for Homenets.

5.1. IS-IS

The only required configuration for IS-IS is a unique area/system identifier. The Homenet implementation of IS-IS uses an autoconfiguration extension defined in an Internet Draft [ISIS-AUTOCONF], to set this value.

5.2. Babel

Babel is a fully self-configuring protocol -- the standard implementation of Babel only requires a list of interfaces in order to start routing.

5.3. Discussion

When combined with HNCP, both protocols are able to run in an unadministered network (but see also Section 7).

6. Support for Source-Specific Routing

Source-Specific Routing is a hard requirement for Homenets, as it will allow traffic to be routed to the correct outbound network based on host source address selection. Routing packets to the wrong outbound network could result in packets being dropped due to ISP ingress filtering rules.

Both Babel and IS-IS have extensions for source-specific routing.

6.1. Source-Specific Routing in IS-IS

IS-IS support for source specific routing is implemented with the addition of a sub-TLV to a reachability (prefix) TLV. The implementation assumes that all IS-IS routers have support for the sub-TLV. This assumption is safe to make due to the requirement that all homenet IS-IS routers also use a homenet specific area ID and cleartext password. Mixing in IS-IS routers without support for source specific routing is not supported as it may cause routing loops.

6.2. Source-Specific Routing in Babel

The Source-specific extension to the Babel routing protocol [BABEL-SS] has been implemented for over a year, has been made widely available and has seen a fair amount deployment as part of OpenWRT and CeroWRT. The source-specific code is currently in the process of being merged into the standard Babel implementation, and is scheduled to be included in version 1.6 (planned for March 2015).

Babel’s source-specific extensions were carefully designed so that source-specific and ordinary (non-specific) routers can coexist in a single routing domain, without causing routing loops. However, unless there is a connected backbone of source-specific routers, any non-specific routers present in the Homenet may experience blackholes. Interoperability between plain Babel and Source-Specific Babel is described in detail in Section VI.A of [SS-ROUTING].

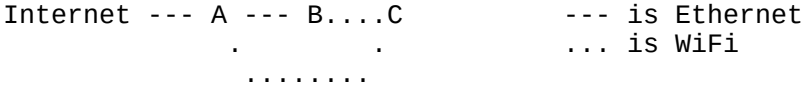
6.3. Discussion

Both protocols have been extended with support for source-specific routing. The IS-IS extension is not able to coexist with non-extended implementations, but this is probably not a serious problem for Homenet.

7. Support for Link Metrics

Typical Homenets are built out of multiple link technologies with very different performance characteristics -- Gigabit Ethernet can easily have three orders of magnitude higher throughput than a marginal wireless link. Both IS-IS and Babel model the notion of greater or lesser desirability of a link by a value called a metric; the smaller the metric, the more desirable the link.

The following example illustrates the importance of assigning suitable metrics to links in a home network:



A is the CPE (edge router), and lives in a storeroom. B is the home's main router, lives in the living room, and is connected to A over Ethernet. C is a wireless router in the guest room, or perhaps in the garden shed. All the routers are equipped with wireless interfaces.

Suppose that the wireless link B..C is solid, but the longer A..C link has very high packet loss. Murphy's law dictates that the link A..C will be just good enough for the routing instances on A and C to become neighbours. If the routing protocol doesn't take link quality into account in its metric computation, even if it is smart enough to prefer wired links to wireless ones, it will prefer the lossy route A..C to the solid route A-B..C.

7.1. Link Metrics in IS-IS

The Homenet implementation of IS-IS uses the wide-metric (24-bit) link metric. Additionally IS-IS includes multi-topology support allowing for a variable number of metrics per link, as well as per-link and per-prefix tags allowing for coloring of links and reachability, and finally per-link and per-prefix sub-tlv's allowing for any future additional extensions.

7.2. Link Metrics in Babel

Since Babel was originally designed for heterogeneous networks, the protocol is able to automatically include a number of sources of information in its metric computation. The current implementation is able to automatically take the following criteria into account:

- whether a link is wired or wireless;

- o packet loss;
- o link latency [BABEL-RTT];
- o intra-route radio interference [BABEL-Z].

This wealth of information can produce conflicting data in edge cases. Babel's loop-avoidance mechanisms ensure that the network remains in a consistent state in all cases, and a hysteresis mechanism ensures that, should a feedback loop occur, the frequency of oscillations remains bounded [DELAY-BASED].

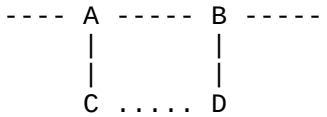
7.3. Discussion

Babel has reasonably good support for dynamically computed metrics for IEEE 802.11 links and high-latency tunnels. The current implementation doesn't have explicit support for variable-speed Ethernet and Powerline Ethernet, both technologies are bundled into a single "good quality link" category.

Current implementations of IS-IS will supply a default metric for links if not configured. We are unaware of any implementation that computes this default metric dynamically, rather it is static and supplied by the vendor. While support for dynamically computed metrics could potentially be added to IS-IS, this is not completely trivial to do right, since a naive approach would cause unacceptable churn, potentially leading to repeated SPF computations and transient microloops. Suitable mitigation techniques could probably be developed, but they would likely be different from the mitigation techniques that work well in Babel, since the link-state algorithm used by IS-IS and the triggered updates used by Babel have fundamentally different dynamics.

8. Support for Stub Networks and Stub Routers

A stub network is one that is attached to a Homenet, possibly through multiple Homenet routers, but must not be used for transit. In the following example, if the dotted link between C and D is a stub network, then it must not be used for transit even if the link between A and B fails:



Similarly a stub router is a router which may advertise and be a destination for stub networks but should never be used for transit traffic.

A number of people have expressed interest in attaching sensor class equipment to Homenets, such as smart thermostats and home automation hardware. Presumably, the sensor-class devices will run over a dedicated low-power link (e.g. IEEE 802.15.4), with a small number of devices acting as gateways between the homenet and the low power network. Ideally, the gateway nodes would not be dedicated devices, but merely sensor nodes that happen to have been equipped with an Ethernet or WiFi interface.

Since low power links have orders of magnitude lower throughput than Homenet links, routing Homenet traffic through the low power link would cause it to collapse. Designating this link as a stub network avoids this issue.

8.1. IS-IS Support for Stub Networks and Stub Routers

In IS-IS reachability (prefixes) and topology (links/adjacencies) are separate things. IS-IS supports stub-networks as defined above simply by advertising the prefix associated with a link, but not the link itself. This is sometimes referred to as a "passive link".

Further an IS-IS router has the ability to set a bit (the overload bit) to indicate that it should not be used for any transit traffic, and that it will only be considered a destination for the prefixes it has advertised, i.e., it is a stub router as defined above. As per the IS-IS standard [ISO10589] an IS-IS router marked as such is not expected to participate in the propagation of link state or the SPF computation.

8.2. Babel Support for Stub Networks

As all distance vector protocols, Babel supports fairly arbitrary route filtering. Designating a stub network is done with two statements in the current implementation’s filtering language.

For resource-limited deployments, a minimalistic, stub-only implementation of Babel is available that consists of less than 1000 lines of C code that compile to a dozen kilobytes of text. Just like the standard implementation, the stub implementation is mostly portable code, and should be easily ported to any embedded environment that supports the "sockets" API.

8.3. Discussion

A tiny, stub-only implementation of Babel is available.

A tiny, stub-router-only implementation of IS-IS is available. This is the "tinyisis" referred to later in the document when comparing implementation sizes.

9. Support for non-IEEE 802.2 link layers

Most link technologies employed in a Homenet use the IEEE 802.2 frame format; this is notably the case of Ethernet, IEEE 802.11 and common powerline technologies. However, other framing formats exist, and at least PPP, PPPoE, IEEE 802.15.4, GRE and various VPN technologies are likely to be used in Homenets.

9.1. IS-IS

IS-IS is built directly above the link layer (IS-IS control data is not encapsulated within IP packets), and therefore needs explicit support for every type of lower layer encapsulation. It is not known what particular kinds of framing the Homenet implementation of IS-IS supports.

9.2. Babel

Babel is built above UDP over link-local IPv6, and is able to run with no modifications over any link that supports IPv6.

10. Security Features

10.1. Security Features in IS-IS

IS-IS offers multiple levels of security from none, to simple clear-text (password) authentication, to fully generic cryptographic authentication using any number of hashing algorithms [RFC5304]. Currently, the Homenet implementation of IS-IS uses the cleartext password set to a predefined value for auto-configuration purposes.

10.2. Security Features in Babel

Babel supports symmetric key authentication using an extensible HMAC-based cryptographic authentication mechanism [RFC7298]. This mechanism is not vulnerable to replay attacks.

10.3. Discussion

Both protocols support password authentication. This probably provides the necessary hooks for the Homenet Configuration Protocol (HNCP) to implement whatever security policies are required by Homenet.

11. Support for Multicast

Although the Homenet WG has not yet determined whether to support multicast in Homenet Networks, it might be desirable to pick a routing protocol that supports multicast, so that it will be easier to add multicast support in the future.

11.1. Multicast Routing in IS-IS

The IS-IS protocol supports multicast routing. However, none of the available implementations include support for multicast.

11.2. Multicast Routing in Babel

There is no support for multicast routing in Babel.

11.3. Discussion

Some experts believe that it may be better to run a dedicated multicast routing protocol rather than extensions to a unicast routing protocol.

12. Implementation Status

There are Homenet implementations of both IS-IS and Babel.

The Homenet implementation of IS-IS is the only IS-IS implementation that supports source-specific routing, which is a hard requirement for Homenet. If source-specific routing is not required, there are several independent, interoperable proprietary implementations of IS-IS (all major router vendors implement IS-IS). We are not aware of any production-quality open-source implementation of IS-IS other than the Homenet one.

There are multiple open source implementations of Babel, two of which support source-specific routing. All implementations (except the stub-only version) were originally derived from the same codebase.

13. Code and State Size

13.1. IS-IS Code and State Size

The Homenet implementation of IS-IS consists of 7000 lines of Erlang code and has an installed size of over 11MB. Its initial memory usage (as reported by the operating system) is 22MB, and its working set increases by XXX bytes for each new edge in the network graph. To put these numbers into perspective, in a network with XXX nodes each of which has XXX neighbours, the Homenet implementation of IS-IS requires XXX bytes for its data structures.

The code size of IS-IS depends greatly on what aspects of the protocol have been implemented. IS-IS supports multiple address families as well as completely different protocol stacks (OSI and IP), multiple area hierachical operation with automatic virtual link support for repairing area partitions, and multiple link types. Additionally many other protocol features have been added over time to augment the protocol or replace previous behavior. The protocol lends itself well to not only extension, but pairing down of features.

For Homenet we need a level-1 only implementation supporting a common topology for IPv4 and IPv6 over broadcast (i.e., ethernet) link types. Additionally, we only require support of the latest extended metric TLVs (i.e., not implement legacy metric support).

[Does that mean that we don’t care about PPP, PPPeE, GRE etc?]

The operational state required by IS-IS is proportional to the number of routers, links, and prefixes in the network. Each router in the network generates and advertises a Link State Protocol Data Unit (LSP) that describes it’s attached links and prefixes. A copy of each of these LSPs is stored by each router in the network. IS-IS uses these LSPs to construct a shortest-path-first (SPF) tree with attached prefix information from which routes to the prefixes are created.

Concrete numbers lacking.

13.2. Babel Code and State Size

The source-specific implementation of Babel, which implements many non-Homenet extensions to the protocol, consists of roughly 10000 lines of C and has an installed size of less than 130kB on AMD-64. Its initial memory usage (as reported by the operating system) is 300kB.

The amount of state stored by a Babel router is at worst one routing table entry for each destination through each neighbour. In the source-specific implementation, one routing entry occupies roughly 100 bytes of memory. To put these figures into perspective, in a network with 1000 nodes, a Babel router with 10 neighbours needs roughly a megabyte of memory to store its routing table (not counting malloc overhead).

The stub-only implementation of Babel consists of 900 lines of C and compiles to 13kB (dynamically linked). Its memory usage (as reported by the operating system) is 200kB, and remains constant (it doesn't perform any dynamic memory allocation).

The stub-only implementation of IS-IS consists of 1300 lines of C and compiles to 18kB (stripped and dynamically linked). Its base memory usage (as reported by 32-bit linux) is 330kB.

13.3. Comparison

Table 1 summarises the sizes of the available Homenet routing protocol implementations. (Babel data courtesy of Steven Barth and Markus Stenberg.)

	babeld (SS)	sbabeld (stub)	AutoISIS (SS)	tinyisis (stub)
Version Date	2598774 2014-09-08	cc7d681 2014-11-21	[update]0.8.0 2014-08-26	3ed2068 2015-02-22
License	MIT	MIT	Apache 2.0	Apache 2.0
Lines of Code	10000 (C)	1000 (C)	7000 (Erlang)	1300 (C)
Installed size (AMD64)	129kB	13kB	732kB	17kB
Total installed size	129kB	13kB	7MB	17kB
Baseline RSS	~300kB	~200kB	11MB	330kB

Table 1: Comparison of Homenet implementation size

In this table, "Installed size" is the size reported by the package manager for the routing daemon's package(s), while "Total installed size" is the sum of the size of the deamon's packages and all its dependencies, excluding the C library.

The erlang AutoISIS has not been optimized for space or features (i.e., it is a full IS-IS multilevel multi-address family implementation). One example of how this affects the reported numbers, is that the erlang logging package is taking up 4MB of ram.

Additionally, as erlang is interpreted and not compiled as with the C implementations, there's not much use in directly comparing the numbers themselves.

[We should add the size of the native Quagga isisd. While this implementation is incomplete and doesn't support the Homenet extensions, it should give us an idea how much we can hope to scale IS-IS down. It's roughly 20000 lines of C, not counting the additional 20000 for zebra. -- jch]

14. Ease of porting

If successful, the Homenet protocols will be implemented on exotic devices, such as sensor-class devices, set-top boxes and mobile phones. Rather than a full Unix system, such devices sometimes provide a more exotic environment, such as an embedded operating system or one designed for mobile phones.

14.1. IS-IS ease of porting

As IS-IS runs directly over layer 2, an implementation needs to be able to send and receive layer 2 frames itself. On Unix systems, this can be done using raw sockets or by hooking into the Berkeley Packet Filter. Such interfaces might not be available on non-Unix systems, and even on Unix are restricted to priviledged processes.

14.2. Babel ease of porting

Babel is layered above UDP. Except for a thin layer interfacing to the kernel, the reference implementation of Babel consists of portable code using the familiar "sockets" API (RFC 3493). Enabling forwarding and manipulating the kernel's routing tables is the only priviledged operation it performs.

Babel has been successfully ported to Android. Android does not currently allow unpriviledged code to manipulate the kernel's routing table, so Babel can only run on "rooted" phones. Should a future version of Android provide the ability to manipulate the kernel's routing table, Babel could conceivably be packaged as an installable "app".

14.3. Discussion

IS-IS is somewhat difficult to port, due to the requirement for raw sockets. Except for the requirement to install routes, Babel is portable code, and has been successfully ported to Android.

15. Behaviour upon resource exhaustion

15.1. IS-IS

The IS-IS protocol has an overload indicator. When the protocol is unable to acquire a needed resource, it enters overloaded state. This state is signaled to the other routers and the overloaded router is considered to be destination only (i.e., it becomes a stub router).

15.2. Babel

Babel degrades gracefully. Since Babel is a distance-vector protocol, a Babel implementation can simply drop excess routes when it runs out of memory. As long as the default route is not discarded, connectivity will be degraded but not completely lost.

The current implementation of Babel discards redundant routes before it starts dropping announcements, but doesn’t yet prioritise the default route. (This behaviour was tested when somebody redistributed the entire IPv6 default-free zone into a Babel network. We had a lot of fun that day.)

16. Performance and scaling on IEEE 802.11 Wireless Networks

We expect wireless networks, and notably IEEE 802.11 wireless networks, to be in wide use in Homenets, not only for client connections but also for router-to-router connections. In fact, some of us believe that the ability to cheaply and easily extend wireless coverage by adding a wireless router is a "killer feature" of Homenet, one that is easy to explain both to one’s boss and to end users.

IEEE 802.11 has some rather unpleasant performance characteristics. First, wireless links are of very variable quality. Second, multicast traffic is sent at a lowest common denominator speed, typically 2Mbit/s, which makes multicast outrageously expensive (13ms for a full-size frame). Third, multicast traffic is not protected by link-layer ARQ, which makes multicast packet drops very common, especially in the presence of collisions, which are particularly likely when the routing protocol is in the process of reconverging.

This has some consequences for routing protocols:

- the routing protocol must be able to deal with varying link qualities (see Section 7);
- if the routing protocol uses multicast for transmitting control data, it must deal gracefully with packet loss during reconvergence;
- the routing protocol must avoid sending large bursts of multicast traffic from multiple nodes simultaneously during reconvergence.

IEEE 802.11 has two main modes of operation. In "infrastructure mode", a small number of "access points" (APs), often just one, communicate with a number of "stations" (STAs); communication between two stations transits through one or at most two APs. In "IBSS mode" (also called "ad hoc mode"), communication is unrestricted, and any node can directly communicate with any other node in range; as there is no link-layer forwarding, IBSS mode does not guarantee that communication is transitive. Virtually all home network deployments today use infrastructure mode, with the router acting as AP and client devices acting as stations. We believe that IBSS may be out of scope for Homenet, but we expect that people will attempt to use the Homenet protocols in IBSS mode, whether we like it or not.

16.1. IS-IS Performance on 802.11

IS-IS is in active use in in the Internet in large non-hierachical (i.e., level-2 or single area level-1) deployments with hundreds of nodes. The protocol has proven to be very scalable.

We are not aware of any results in the open litterature about the behaviour of IS-IS over IEEE 802.11. In particular, we do not know how IS-IS’s reliable flooding mechanism behaves over IEEE 802.11.

16.2. Babel Performance on 802.11

Babel has been carefully optimised for IEEE 802.11 networks. While Babel transmits most control data over multicast, it applies large amounts of random jitter (on the order of seconds) to all but its most urgent control messages. Roughly speaking, Babel sends in a timely manner those control messages that are likely to repair a broken route, but delays sending those messages that merely announce a slightly better route than one that is already available. This mechanism has been shown to work well in dense wireless deployments, with recent versions of Babel being able to avoid link-layer meltdowns even when a whole network is being rebooted simultaneously.

As explained in Section 7, Babel performs link quality estimation on wireless links in a manner that works relatively well with the 802.11 MAC. In addition, Babel has provisions for estimating radio interference [BABEL-Z], which is necessary in order to provide decent throughput on multi-hop radio routes.

Babel has been designed to work well in IBSS mode, where communication is not transitive and therefore a packet may be routed through the same interface it was received on.

16.3. Discussion

Babel has been designed with wireless networks in mind, and has been shown to work reasonably well even in the extremely hostile environment of wireless mesh networks built over IEEE 802.11 in IBSS ("ad hoc") mode.

We are not aware of any results about the behaviour of IS-IS's reliable flooding sub-protocol over IEEE 802.11 multicast. IS-IS will not work over IEEE 802.11 IBSS mode without changes, since both the pseudonode optimisation and DIS election assume that communication is transitive.

17. Standardization Status

17.1. IS-IS Standardization

IS-IS is an ISO Standard documented in ISO/IEC 10589:2002 [ISO10589] There is an active IETF IS-IS Working Group (isis-wg) that maintains and extends the IS-IS protocol, and the IS-IS protocol has been extended in several IS-IS Working Group documents.

The autoconfiguration and source-specific extensions to IS-IS, which are both hard requirements for Homenet, are documented in (non-WG) Internet Drafts [ISIS-AUTOCONF] [ISIS-SS].

17.2. Babel Standardization Status

Babel is documented in an Experimental RFC (RFC 6126) published in 2011, and it has been updated in several individual-submission RFCs and Internet Drafts. An Internet Draft establishing an IANA registry of Babel extensions has been submitted for publication as an RFC [BABEL-EXT].

The use of Babel in a Standards Track Homenet RFC would require a "downref" to non-Standards Track documents. It would also be necessary to finish publishing the extensions that are needed for the Homenet use case as RFCs.

18. Evaluation of RFC 7368 Criteria

Section 3.5 of [RFC7368] lists a set of criteria that the Homenet routing protocol must satisfy.

- border discovery: out of scope, as this is done by HNCP.
- self configuring: both protocols are self-configuring, see Section 5.
- behaviour during reconfiguration and reconvergence:
  - both protocols are able to establish adjacencies and to route IPv6 before they even receive an IPv6 address due to their use of stable neighbour identifiers;
  - Babel will avoid creating loops even during reconvergence, but might create temporary blackholes;
  - IS-IS may create short-lived loops during reconvergence;
  - since HNCP doesn’t depend on unicast, in principle it is not impacted by the routing protocol’s behaviour during reconvergence; however, on a wireless network, the interference caused by routing loops may delay HNCP’s reconvergence.
- the Homenet routing protocol should be based on a previously deployed protocol:
  - IS-IS is more widely deployed in carrier networks,
  - there is more experience with running Babel on Plastic Home Routers
  - Only the Babel implementation of source-specific routing has seen any deployment.
- support for IPv4: both protocols are intrinsically double-stack -- a single routing instance is able to carry both IPv6 and IPv4.
- multiple types of physical interfaces must be accounted for: only Babel is able to differentiate between different link technologies, see Section 7; nothing is known about IS-IS’s behaviour on wireless links.
- minimising convergence time: both protocols converge fast, see Section 4.

- support the generic use of multiple Internet connections: both protocols support source-specific routing, see Section 6.
- scalable to higher hop counts: both protocols have reasonably wide metrics (24 bits in IS-IS, 16 bits in Babel).
- support for attached LLNs and other non-transit networks: both protocols are able to designate a link as non-transit. Tiny stub implementations of Babel and IS-IS are available. See Section 8.
- support for Multicast: IS-IS is able to carry multicast routes, Babel is not.

19. Evaluation of RFC 5218 Criteria

19.1. Critical Success Factors

Does the protocol exhibit one or more of the critical initial success factors as defined in RFC 5218?

19.1.1. IS-IS Success Factors

IS-IS exhibits the following critical initial success factors:

- Positive Net Value:
  - Hardware cost: None.
  - Operational interface: Existing and extensive.
  - Retraining: None.
  - Business dependencies: None.
- Incremental Deployment: Yes.
- Open Code Availability: Yes. One implementation of the Homenet extensions, multiple proprietary implementations of the base protocol.
- Freedom from Usage Restrictions: Yes.
- Open Specification Availability: Yes.
- Open Maintenance Processes: Yes.

- Good Technical Design: Proven with extensive deployment and experience with the base protocol, little deployment of the Homenet extensions.
- Extensible: Yes.
- No Hard Scalability bound: Yes.
- Threats Sufficiently Mitigated: Yes.

19.1.2. Babel Success Factors

Babel exhibits the following critical initial success factors:

- Positive Net Value:
  - Hardware cost: None.
  - Operational interface: tcpdump and wireshark support, dedicated monitoring software.
  - Retraining: None.
  - Business dependencies: None.
- Incremental Deployment: Yes.
- Open Code Availability: Yes. Multiple implementations derived from a common source.
- Freedom from Usage Restrictions: Yes.
- Open Specification Availability: Yes.
- Open Maintenance Processes: IANA registry in the process of being created.
- Good Technical Design: Yes.
- Extensible: Yes.
- No Hard Scalability bound: Yes.
- Threats Sufficiently Mitigated: probably.

19.2. Willing Implementors

Are there implementers who are ready to implement the technology in ways that are likely to be deployed?

19.2.1. IS-IS

There is only one implementation of autoconfiguration and source-specific routing for IS-IS. There are some other open source implementations of the base protocol, but they are incomplete (as of February 2015).

As all major routing vendors have (proprietary) IS-IS implementations, the barrier for implmeneting IS-IS for Homenet use is probably manageable, assuming that the willingness to implement modifications needed for Homenet use is present.

19.2.2. Babel

The Babel implementation is open source software (MIT licensed), and the codebase has proven of sufficiently high quality to be easily extended by people who were not in direct contact with the author [RFC7298].

With the exception of a small number of functions used for interfacing with the kernel’s routing tables, Babel is portable code, and is easily ported to any environment that supports the familiar "sockets" API.

19.3. Willing Customers

Are there customers (especially high-profile customers) who are ready to deploy the technology?

19.3.1. IS-IS

Yes. IS-IS is already widely deployed in operational networks.

19.3.2. Babel

Source-Specific Babel is currently deployed as part of the OpenWRT and CeroWRT operating systems. Additionally, the current version is used as a testbed for the Homenet configuration protocol.

19.4. Potential Niches

Are there potential niches where the technology is compelling?

19.4.1. IS-IS

[TBD]

19.4.2. Babel

Babel is a simple and flexible routing protocol. Like most distance-vector protocols, it requires little to no configuration in most topologies, and has proved popular in scenarios where competent network administration was not available. In addition, it has been shown to be particularly useful in scenarios where non-standard dynamically computed metrics are beneficial, notably wireless mesh networks and overlay networks.

19.5. Complexity Removal

If so, can complexity be removed to reduce cost?

19.5.1. IS-IS

As mentioned previously IS-IS can be significantly and easily pared down to fit the more limited scope of homenet use. However, no such pared down implementation exists, and the subset of the protocol that needs to be implemented has never been formally defined.

19.5.2. Babel

Babel is a fairly simple protocol -- RFC 6126 is just 40 pages long (not counting informative appendices), and it has been successfully explained to fourth year university students in less than two hours.

The stub-only implementation of Babel consists of 900 lines of C code, and has deliberately been kept as simple as possible. We expect a competent engineer to get up to speed with it within hours.

19.6. Killer App

Is there a potential killer app? Or can the technology work underneath existing unmodified applications?

19.6.1. IS-IS

As IS-IS already qualifies as successful (bordering on wildly) a killer app is not particularly relevant.

19.6.2. Babel

Since Babel requires virtually no configuration, it is particularly suitable to scenarios where a dedicated network administrator is not available. Additionally, its support for dynamically computed non-standard metrics makes it particularly appealing in highly heterogeneous networks, (networks built on multiple link-layer technologies with widely varying performance characteristics).

19.7. Extensible

Is the protocol sufficiently extensible to allow potential deficiencies to be addressed in the future?

19.7.1. IS-IS

IS-IS has been shown to be incredibly extensible, originally designed for a completely different protocol stack (OSI) it was easily adapted for IP use, then to multiple address families (IPv4, IPv6) and multi-topology. Indeed one of the major drivers of IS-IS's success is its extensibility and adaptability.

19.7.2. Babel

The extension mechanisms built into the Babel protocol [BABEL-EXT] have been used to build a number of extensions, including one that significantly extends the structure of announcements [BABEL-SS] and one that needs a non-trivial extension to the space of metrics [BABEL-Z].

All Babel extensions that have been defined interoperate with the original protocol, as defined in RFC 6126, to the extent possible. For example, a router that doesn't implement the radio interference extension will just ignore the frequency information attached to route updates, which may lead to slightly sub-optimal routing but will cause neither blackholes nor routing loops. To the contrary, a router that doesn't support the source-specific extensions will silently ignore any source-specific update, and therefore route according to the non-specific subset of available routes, which might cause blackholes, but under no circumstances will cause routing loops. Backwards compatibility provisions are described in Section 4 of [BABEL-EXT].

19.8. Success Predictable

If it is not known whether the protocol will be successful, should the market decide first? Or should the IETF work on multiple alternatives and let the market decide among them? Are there factors listed in this document that may predict which is more likely to succeed?

19.8.1. IS-IS

For IS-IS the market has already decided that the protocol is successful in a fairly wide variety of deployments.

19.8.2. Babel

Plain (non-source-specific) Babel has seen a modest amount of deployment, most notably for routing over wireless mesh networks and intercontinental overlay networks. Babel is included as an installable package in many Linux distributions, which makes it easily available to interested parties.

Source-specific Babel is probably the only source-specific routing protocol that has seen any deployment. Source-specific Babel is available as an installable package for OpenWRT, and is used by default by CeroWRT.

20. Acknowledgments

The authors are grateful for the input of Steven Barth, Denis Ovsienko, Mark Townsley, and Curtis Villamizar.

21. Informative References

[BABEL-EXT]
Chroboczek, J., "Extension Mechanism for the Babel Routing Protocol", draft-chroboczek-babel-extension-mechanism-03 (work in progress), June 2013.

[BABEL-RTT]
Jonglez, B. and J. Chroboczek, "Delay-based Metric Extension for the Babel Routing Protocol", Internet Draft draft-jonglez-babel-rtt-extension-00, July 2014.

[BABEL-SS]
Boutier, M. and J. Chroboczek, "Source-Specific Routing in Babel", draft-boutier-babel-source-specific-00 (work in progress), November 2014.

[BABEL-Z] Chroboczek, J., "Diversity Routing for the Babel Routing Protocol", draft-chroboczek-babel-diversity-routing-00 (work in progress), July 2014.

[DELAY-BASED] Jonglez, B. and M. Boutier, "A delay-based routing metric", March 2014, <http://arxiv.org/abs/1403.3488>.

[ISIS-AUTOCONF] Liu, B., "ISIS Auto-Configuration", draft-liu-isis-auto-conf-03 (work in progress), October 2014.

[ISIS-SS] Baker, F. and D. Lamparter, "IPv6 Source/Destination Routing using IS-IS", draft-baker-ipv6-isis-dst-src-routing-02 (work in progress), October 2014.

[ISO10589] ISO/IEC, "Intermediate system to Intermediate system intra-domain routeing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode Network Service (ISO 8473) Second Edition", Novmeber 2002.

ISO 10589:2002 Second Edition

[RFC1195] Callon, R., "Use of OSI IS-IS for routing in TCP/IP and dual environments", RFC 1195, December 1990.

[RFC5304] Li, T. and R. Atkinson, "IS-IS Cryptographic Authentication", RFC 5304, October 2008.

[RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, October 2008.

[RFC5308] Hopps, C., "Routing IPv6 with IS-IS", RFC 5308, October 2008.

[RFC6126] Chroboczek, J., "The Babel Routing Protocol", RFC 6126, April 2011.

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

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

[SS-ROUTING]
Boutier, M. and J. Chroboczek, "Source-sensitive routing", December 2014, <http://arxiv.org/abs/1403.0445>.

Authors' Addresses

Margaret Wasserman
Painless Security
356 Abbott Street
North Andover, MA 01845
USA

Phone: +1 781 405-7464
Email: mrw@painless-security.com
URI: http://www.painless-security.com

Christian E. Hopps
Deutsche Telekom

Email: chopps@chopps.org

Juliusz Chroboczek
University of Paris-Diderot (Paris 7)

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