

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 5, 2015

X. Bonnetain
Cisco Systems
July 4, 2014

HNCP security based on routers trust
draft-bonnetain-hncp-security-00

Abstract

This memo describes how to secure HNCP. Based on asymmetric cryptography and trust relationships, it ensures integrity, authenticity and, to a lesser extent, secrecy and non-repudiation. This secrecy can be used to encrypt part of the shared data, or to secure other protocols, like the routing protocol.

Status of This Memo

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

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

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

This Internet-Draft will expire on January 5, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	2
2. Requirements language	3
3. HNCP message layout	3
4. Public/private key pair	4
4.1. Node Key TLV	4
5. Signature	4
5.1. Signature TLV	5
6. Trust relationships	5
6.1. Trusted nodes set	5
6.2. Relaying Node Data TLVs	6
6.3. Using Node Data TLVs	6
6.4. Trust Link TLV	6
7. Symmetric encryption	6
7.1. Encryption functioning	6
7.2. Shared key management	7
7.3. Symmetric cipher TLVs	8
7.3.1. Shared key TLV	8
7.3.2. Private Data TLV	8
8. Trust establishment & revocation	9
8.1. Trust relationship establishment	9
8.1.1. Node Information TLV	10
8.1.2. Simple Trust Establishment	10
8.2. Trust revocation	11
9. Security Considerations	11
10. References	11
10.1. Normative References	11
10.2. Informative References	12
Appendix A. Acknowledgments	12
Author's Address	12

1. Introduction

Current home networks, left unsecured, can be subject to various attacks: IP spoofing, Router Advertisement spoofing... Besides, HNCP can extend those threats to the whole network and allow any router to impact any part of it.

Protection against threats affecting a single link is out of the scope of this document. Instead, it intends to prevent these threats from being extended to multiple links. In particular, an attacker connected to an unsecured link shouldn't be able to cause any harm on a secured link.

This memo proposes a distributed approach to establish and use secured relationships between routers.

The rest of this memo is organized as follow. Section 2 defines the usual keywords, Section 3 overviews the data structure changes, Section 4 describes the public/private key pair use, Section 5 shows how signatures are done, Section 6 describes the trust mechanism, Section 7 describes how to encrypt data with trusted peers and Section 8 describes how trust is established or revoked.

2. Requirements language

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

3. HNCP message layout

This memo defines a new layout for the Node Data TLV (defined in [I-D.ietf-homenet-hncp]): all data is signed, and part of it can be encrypted in a dedicated container, the Private Data TLV (Section 7.3.2).

Node Data TLV model:

```
+ Node Data TLV
|
+---+ Identity & Trust information
|
+---+ Cryptography elements
|
+---+ Other public TLVs
|
+---+ Private Data TLV
|   |
|   + Encrypted TLVs
|
+---+ Signature
```

The following TLVs MUST NOT be encrypted:

- o Node Key TLV (Section 4.1)
- o Trust Link TLVs (Section 6.4)
- o Shared Key TLVs (Section 7.3.1)
- o Node Information TLV (Section 8.1.1)
- o Signature TLV (Section 5.1)

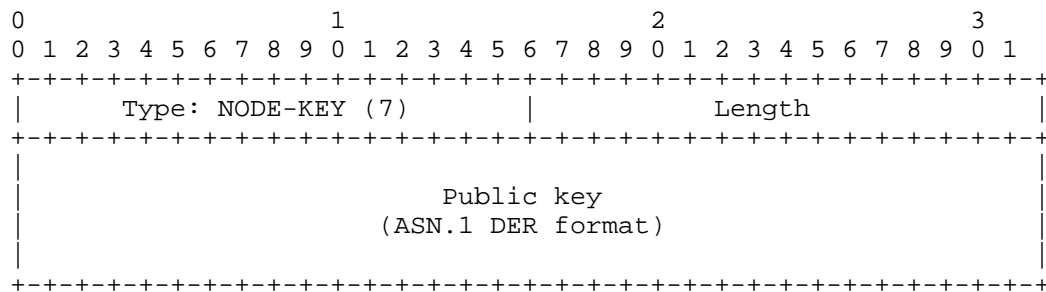
The Node Data TLV MUST contain a Node Key TLV and a Signature TLV.

4. Public/private key pair

HNCP security is based on asymmetric cryptography. Hence, all HNCP nodes MUST have their own public/private key pair, and they MUST advertise one and only one Node Key TLV containing the public key, included in their Node Data TLV defined in [I-D.ietf-homenet-hncp].

Furthermore, all HNCP nodes MUST use their public key as node identifiers, which implies that the node identifier hash used in HNCP MUST be the MD5 hash of the public key. As MD5 is not collision resistant, other hash functions could be more suitable for that purpose.

4.1. Node Key TLV



Public key: The public key used for node identification and signature verification. It is an ASN.1 SubjectPublicKeyInfo field, as defined in [RFC3280], section 4.1. ASN.1 Distinguished Encoding Rules (DER) are used to encode it to ensure the uniqueness of the hash of the public key. See [CCITT.X690.2002] for details on ASN.1 and DER. This field is the input to generate the node identifier hash.

5. Signature

Any Node Data TLV MUST be signed with the originator's private key. The input of the signature function is the concatenation of:

- o The HNCP update sequence number (See [I-D.ietf-homenet-hncp], section 5.2.4).
- o The complete set of TLVs embedded in the Node Data TLV, not including the Signature TLV itself or the others elements of the Node Data TLV header.

The signature is then included in a Signature TLV, as the last Node Data TLV's sub-TLV.

5.1. Signature TLV

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|  Type: SIGNATURE (65535)      |      Length: >= 12      |
+-----+-----+-----+-----+-----+-----+-----+-----+
|      Signature algorithm      |                           |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                               |                           |
|      Signature of the preceding TLVs      |               |
|                               |                           |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Signature algorithm: The algorithm used for signature (including padding and hash function). MUST be compatible with the published public key.

Signature: The raw signature, verifiable with the originator's public key

The signature parameters could be a new IANA register or an existing ones. Input from the working group is required to decide which default signature scheme should be used.

6. Trust relationships

The signature process specified previously ensures data integrity and authenticity. In this document, we propose to enforce authorization using directed trust relationships.

6.1. Trusted nodes set

Each node MUST maintain a list of trusted nodes identifier (public keys, see Section 4) and advertise the hashes of these identifiers as part of the Node Data TLV using Trust Link TLVs. Sharing this information allows each node to have a local copy of all the trust relationships.

Using this database, each node can create a graph representing the trust network, where trust relationships are oriented edges.

Each node can then compute the Trusted Nodes Set, containing all reachable nodes with a trust path from the local node to them, and vice versa.

This definition ensures that, given a common view of the trust graph, all computed Trusted Nodes Sets are either equal or disjoint.

6.2. Relaying Node Data TLVs

All HNCP updates with an invalid signature or an invalid node identifier hash MUST be ignored.

All other updates must be relayed, even those from nodes that are not included in the Trusted Nodes Set, but such updates SHOULD be rate and size limited. This way, any node can compute the Trusted Nodes Set based on a complete knowledge of the trust network, which is necessary in some cases.

6.3. Using Node Data TLVs

All Node Data TLVs' authenticity must be verified using the Node Data TLV's update number, the Node Key TLV and the Signature TLV, as specified in Section 5.1.

TLVs contained in Node Data TLVs originated by nodes that are not in the Trusted Node Set must be ignored, except for the TLVs mandatory to establish the trust.

6.4. Trust Link TLV

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|      Type: TRUST-LINK (70)      |      Length: 24      |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     H(node identifier)                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

This TLV contains the hash of the public key of a trusted node.

7. Symmetric encryption

7.1. Encryption functioning

Asymmetric cryptography used in previous sections ensures integrity, but not secrecy. HNCP security is hence done in two steps: asymmetric cryptography is used to encrypt a symmetric shared key, which is then used to encrypt data. Most of the data can be encrypted this way, but it isn't required. For example, to secure

the home, the routing protocol (unicast or multicast) must be secured as well, and a dedicated shared secret key can be flooded to the Trusted Nodes Set using this procedure.

Shared secret establishment works as follow:

- o A router generates a reasonably long secret symmetric key, using suitable random source, which will be used only in HNCP.
- o For each router in the Trusted Nodes Set, the originator creates a Shared Key TLV (Section 7.3.1), with the secret key encrypted with the trusted router's public key.
- o Each router in the Trusted Nodes Set can decrypt the key using its private key.
- o Secret data is encrypted symmetrically with the Shared Key and put in the Private Data TLV (Section 7.3.2).

Although a single key is enough, multiple keys may coexist. Keys are therefore identified by the couple (key identifier, key emitter).

7.2. Shared key management

A node with an empty Trusted Node Set doesn't need to share its private data and therefore doesn't have to generate a shared key.

When the local node see some other nodes in its Trusted Nodes Set, two situations can occur:

- o One of the nodes in the Trusted Nodes Set already advertises a Shared Key. The local node MUST wait for the shared key emitter to encrypt the key for him.
- o No shared key is advertised. The node in the Trusted Nodes Set with the highest Node Identifier Hash MUST generate a key and advertise it to the Trusted Nodes Set by sending one Shared Key TLV (Section 7.3.1) per other node in the Trusted Node Set.

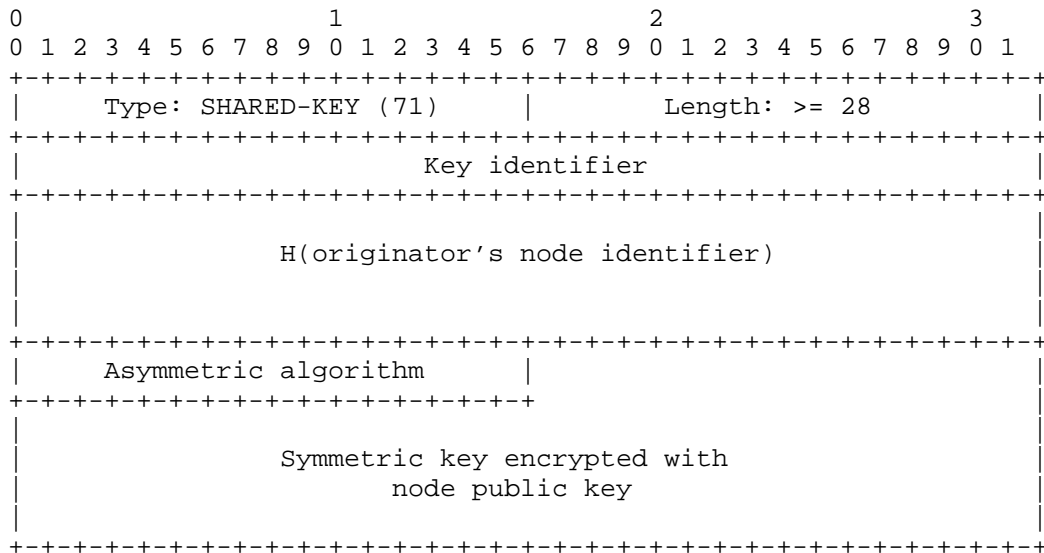
If more than one key is available for a node, it SHOULD use the key of the emitter with the highest node identifier hash, and other key emitter SHOULD stop advertising their key.

If a key is no longer advertised, nodes MUST stop using it.

If a node quits the Trusted Node Set (in case of trust revocation (Section 8.2)), all keys and all the secrets shared with it MUST be renewed.

7.3. Symmetric cipher TLVs

7.3.1. Shared key TLV



Key identifier: The identifier of the published key. A router SHOULD NOT publish a new key if a usable and non-compromised key can be used instead.

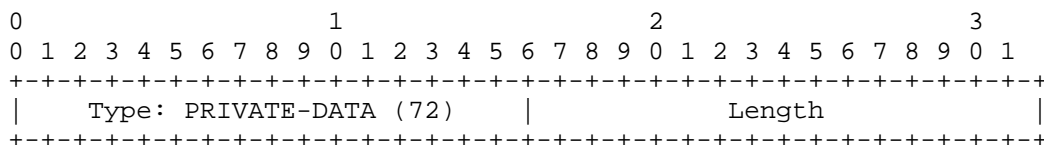
Hash: The identifier of the node that is able to decipher the encrypted secret key.

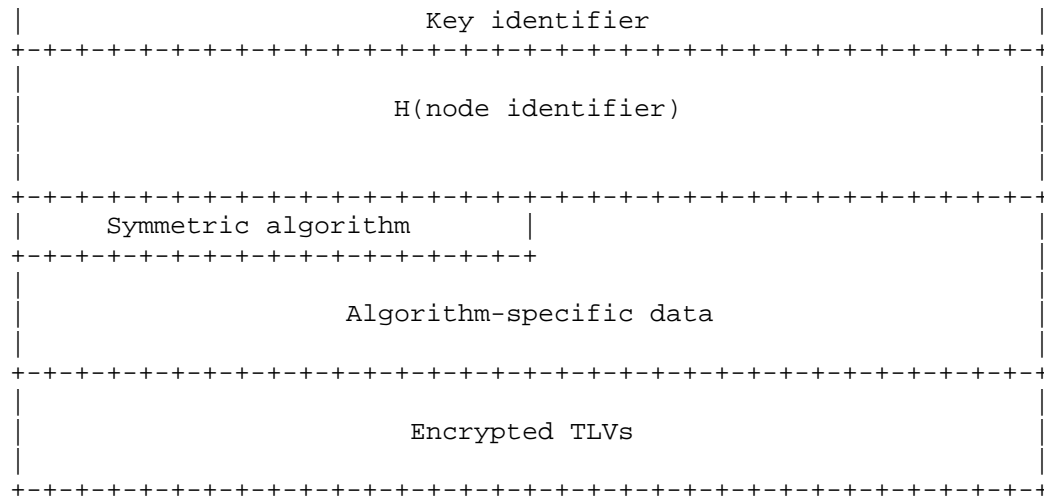
Asymmetric algorithm: The algorithm used for encryption

Symmetric key: The raw shared key, encrypted with the target node public key.

As for signature specification, the algorithm list can be a new IANA registry, and contains various standardized asymmetric encryption algorithms (such as RSAES-OAEP or RSAES-PKCS1-v1.5, defined in [RFC3447]).

7.3.2. Private Data TLV





This TLV contains the information to choose and use the shared key.

Key identifier: The id of the used key. Matches Shared Key TLV (See above)

Hash: The node identifier hash of the emitter of the key.

Symmetric algorithm: The encryption algorithm

Algorithm-specific data: Everything needed to decrypt, but the key. Contains the initialization vector, if any.

Encrypted TLVs: A blob of encrypted data.

8. Trust establishment & revocation

8.1. Trust relationship establishment

Using HNCP, a node can obtain the public keys of connected nodes and decide whether to trust them or not. The way trust relationships can be created is implementation specific, and therefore out of the scope of this document. A way to establish trust with a minimal user input is described in Section 8.1.2.

For instance, the following methods may be used:

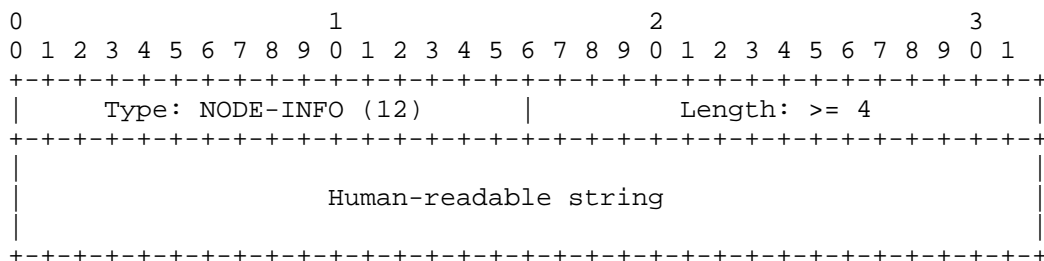
- o Preinstalled public key files
- o Automatic trust of nodes on a particular network interface

- o Public keys fetched by another protocol
- o Managed by the user through a web interface, helped with gathered information
- o User action like pressing button that temporarily enables trust establishment
- o Centralized trust bootstrapping as specified in [I-D.behringer-homenet-trust-bootstrap]
- o Trust link management from a remote server

Such a process should be as user-friendly as possible. Implementations should try to fit to existing security processes like pushing buttons, entering pin codes or relying on certificates in order to provide a satisfactory security level.

Nodes may also look at the trust network to suggest to the user which trust relationship should be established.

8.1.1. Node Information TLV



This TLV contains information about the node, (such as a serial number, a manufacturer, a name...) in a human-readable way, to help a user to choose between nodes. It can be used to help a user to identify and choose to-be-trusted nodes. However, this information on mistrusted nodes can't be verified, is not fully reliable, and can be the same for different nodes.

8.1.2. Simple Trust Establishment

This method allows a router to trust another router with a minimal user involvement. Any action (like pressing a button for a few seconds) can trigger it.

At first, the router trusts all its network neighbors it doesn't already trusts (eventually limited to some interfaces). Then, After

a timeout, only the neighbors that trusts the router back remains trusted.

Trusting only the routers that trusts you back limits the number of trusts links advertised, and allow different homenets to be on the same link. This way of creating trust links should not be used on unsecured links by default (e.g unsecured WiFi, or WiFi if you want to ensure some physical protection).

A router may also automatically trust newcomers on some interfaces. However, this should not be the default behavior.

8.2. Trust revocation

A node advertises all the hashes of the nodes it trusts. A node can revoke a trust link by stopping the advertisement of the according TLV.

9. Security Considerations

This document provides some level of security to HNCP.

10. References

10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3447] Jonsson, J. and B. Kaliski, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1", RFC 3447, February 2003.
- [RFC3280] Housley, R., Polk, W., Ford, W., and D. Solo, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 3280, April 2002.
- [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-13 (work in progress), March 2014.
- [I-D.ietf-homenet-hncp]
Stenberg, M. and S. Barth, "Home Networking Control
Protocol", draft-ietf-homenet-hncp-00 (work in progress),
April 2014.

[CCITT.X690.2002]

International Telephone and Telegraph Consultative Committee, "ASN.1 encoding rules: Specification of basic encoding Rules (BER), Canonical encoding rules (CER) and Distinguished encoding rules (DER)", CCITT Recommendation X.690, July 2002.

10.2. Informative References

[I-D.behringer-homenet-trust-bootstrap]

Behringer, M., Pritikin, M., and S. Bjarnason, "Bootstrapping Trust on a Homenet", draft-behringer-homenet-trust-bootstrap-02 (work in progress), February 2014.

Appendix A. Acknowledgments

We would like to thank all the people who participated in this document. In particular, we would like to thank Mark Townsley, Pierre Pfister, Markus Stenberg and Steven Barth for interesting advice in this problem space.

Author's Address

Xavier Bonnetain
Cisco Systems
Paris
France

Email: xavier.bonnetain_ietf@polytechnique.org

Homenet Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 27, 2014

M. Stenberg
S. Barth
June 25, 2014

Home Networking Control Protocol
draft-ietf-homenet-hncp-01

Abstract

This document describes the Home Networking Control Protocol (HNCP), a minimalist state synchronization protocol for Homenet routers.

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 December 27, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	3
2. Requirements language	3
3. Data model	3
4. Operation	4
4.1. Trickle-Driven Status Updates	4
4.2. Protocol Messages	5
4.2.1. Network State Update (NetState)	5
4.2.2. Network State Request, (NetState-Req)	5
4.2.3. Node Data Request (Node-Req)	6
4.2.4. Network and Node State Reply (NetNode-Reply)	6
4.3. HNCP Protocol Message Processing	6
4.4. Adding and Removing Neighbors	8
4.5. Purging Unreachable Nodes	8
5. Type-Length-Value objects	8
5.1. Request TLVs (for use within unicast requests)	9
5.1.1. Request Network State TLV	9
5.1.2. Request Node Data TLV	9
5.2. Data TLVs (for use in both multi- and unicast data)	10
5.2.1. Node Link TLV	10
5.2.2. Network State TLV	10
5.2.3. Node State TLV	10
5.2.4. Node Data TLV	11
5.2.5. Node Public Key TLV (within Node Data TLV)	12
5.2.6. Neighbor TLV (within Node Data TLV)	12
5.3. Custom TLV (within/without Node Data TLV)	12
5.4. Version TLV (within Node Data TLV)	13
5.5. Authentication TLVs	13
5.5.1. Certificate-related TLVs	13
5.5.2. Signature TLV	14
6. Border Discovery and Prefix Assignment	14
7. DNS-based Service Discovery	19
7.1. DNS Delegated Zone TLV	19
7.2. Domain Name TLV	19
7.3. Router Name TLV	19
8. Routing support	20
8.1. Protocol Requirements	20
8.2. Announcement	20
8.3. Protocol Selection	21
8.4. Fallback Mechanism	21
9. Security Considerations	22
10. IANA Considerations	23
11. References	24
11.1. Normative references	24
11.2. Informative references	25
11.3. URIs	26

Appendix A. Some Outstanding Issues	26
Appendix B. Some Obvious Questions and Answers	27
Appendix C. Changelog	28
Appendix D. Draft source	28
Appendix E. Acknowledgements	28
Authors' Addresses	28

1. Introduction

HNCP is designed to synchronize state across a Homenet (or other small site) in order to facilitate automated configuration within the site, integration with trusted bootstrapping

[I-D.behringer-homenet-trust-bootstrap] and default perimeter detection [I-D.kline-homenet-default-perimeter], automatic IP prefix distribution [I-D.pfister-homenet-prefix-assignment], and service discovery across multiple links within the homenet as defined in [I-D.stenberg-homenet-dnssd-hybrid-proxy-zeroconf].

HNCP is designed to provide enough information for a routing protocol to operate without homenet-specific extensions. In homenet environments where multiple IPv6 prefixes are present, routing based on source and destination address is necessary

[I-D.troan-homenet-sadr]. Routing protocol requirements for source and destination routing are described in section 3 of [I-D.baker-rtgwg-src-dst-routing-use-cases].

A GPLv2-licensed implementation of the HNCP protocol is currently under development at <https://github.com/sbyx/hnetd/> and the binaries are available in the routing feed of OpenWrt [2] trunk release. Some information how to get started with it is available at [3]. Comments and/or pull requests are welcome.

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. Data model

The data model of the HNCP protocol is simple: Every participating node has (and also knows for every other participating node):

A unique node identifier. It may be a public key, unique hardware ID, or some other unique blob of binary data which HNCP can run a hash upon to obtain a node identifier that is very likely unique among the set of routers in the Homenet.

A set of Type-Length-Value (TLV) data it wants to share with other routers. The set of TLVs have a well-defined order based on ascending binary content that is used to quickly identify changes in the set as they occur.

Latest update sequence number. A 32 bit number that is incremented anytime TLV data changes are detected.

Relative time, in milliseconds, since last publishing of the current TLV data set. It is also 32 bit number on the wire.

If HNCP security is enabled, each node will have a public/private key pair defined. The private key is used to create signatures for messages and node state updates and never sent across the network by HNCP. The public key is used to verify signatures of messages and node state updates.

4. Operation

HNCP is designed to run on UDP port IANA-UDP-PORT, using both link-local scoped IPv6 unicast and link-local scoped IPv6 multicast messages to address IANA-MULTICAST-ADDRESS for transport. The protocol consists of Trickle [RFC6206] driven multicast status messages to indicate changes in shared TLV data, and unicast state synchronization message exchanges when the Trickle state is found to be inconsistent.

4.1. Trickle-Driven Status Updates

Each node MUST send link-local multicast NetState Messages (Section 4.2.1) each time the Trickle algorithm [RFC6206] indicates they should on each link the protocol is active on. When the locally stored network state hash changes (either by a local node event that affects the TLV data, or upon receipt of more recent data from another node), all Trickle instances MUST be reset. Trickle state MUST be maintained separately for each link.

Trickle algorithm has 3 parameters; Imin, Imax and k. Imin and Imax represent minimum and maximum values for I, which is the time interval during which at least k Trickle updates must be seen on a link to prevent local state transmission. Bounds for recommended Trickle values are described below.

k=1 SHOULD be used, as given the timer reset on data updates, retransmissions should handle packet loss.

Imax MUST be at least one minute.

Imin MUST be at least 200 milliseconds (earliest transmissions may occur at $Imin/2 = 100$ milliseconds given minimum values as per the Trickle algorithm).

4.2. Protocol Messages

Protocol messages are encoded as purely as a sequence of TLV objects (Section 5). This section describes which set of TLVs MUST or MAY be present in a given message.

In order to facilitate fast comparing of local state with that in a received message update, all TLVs in every encoding scope (either root level, 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 preamble.

4.2.1. Network State Update (NetState)

This Message SHOULD be sent as a multicast message.

The following TLVs MUST be present at the start of the message:

Node Link TLV (Section 5.2.1).

Network State TLV (Section 5.2.2).

The NetState Message MAY contain Node State TLV(s) (Section 5.2.3). If so, either all Node State TLVs are included (referred to as a "long" NetState Message), or none are included (referred to as a "short" NetState Message). The NetState Message MUST NOT contain only a portion of Node State TLVs as this could cause problems with the Protocol Message Processing (Section 4.3) algorithm. Finally, if the long version of the NetState message would exceed the minimum IPv6 MTU when sent, the short version of the NetState message MUST be used instead.

If HNCP security is enabled, authentication TLVs (Section 5.5) MUST be present.

4.2.2. Network State Request, (NetState-Req)

This Message MUST be sent as a unicast message.

The following TLVs MUST be present at the start of the message:

Node Link TLV (Section 5.2.1).

Request Network State TLV (Section 5.1.1).

If HNCP security is enabled, authentication TLVs (Section 5.5) MUST be present.

4.2.3. Node Data Request (Node-Req)

This Message MUST be sent as a unicast message.

MUST be present:

Node Link TLV (Section 5.2.1).

one or more Request Node Data TLVs (Section 5.1.2).

If HNCP security is enabled, authentication TLVs (Section 5.5) MUST be present.

4.2.4. Network and Node State Reply (NetNode-Reply)

This Message MUST be sent as a unicast message.

MUST be present:

Node Link TLV (Section 5.2.1).

Network State TLV (Section 5.2.2) and Node State TLV (Section 5.2.3) for every known node by the sender, or

one or more combinations of Node State and Node Data TLVs (Section 5.2.4).

If HNCP security is enabled, authentication TLVs (Section 5.5) MUST be present.

4.3. HNCP Protocol Message Processing

The majority of status updates among known nodes are handled via the Trickle-driven updates (Section 4.1). This section describes processing of messages as received, along with associated actions or responses.

HNCP is designed to operate between directly connected neighbors on a shared link using link-local IPv6 addresses. If the source address of a received HNCP packet is not an IPv6 link-local unicast address, the packet SHOULD be dropped. Similarly, if the destination address

is not IPv6 link-local unicast or IPv6 link-local multicast address, packet SHOULD be dropped.

Upon receipt of:

NetState Message (Section 4.2.1): If the network state hash within the message matches the hash of the locally stored network state, consider Trickle state as consistent with no further processing required. If the hashes do not match, consider Trickle state as inconsistent. In this case, if the message is "short" (contains zero Node State TLVs), reply with a NetState-Req Message (Section 4.2.2). If the message was in long format (contained all Node State TLVs), reply with NodeState-Req (Section 4.2.3) for any nodes for which local information is outdated (local update number is lower than that within the message), potentially incorrect (local update number is same and the hash of node data TLV differs) or missing. Note that if local information is more recent than that of the neighbor, there is no need to send a message.

NetState-Req (Section 4.2.2): Provide requested data in a NetNode-Reply Message containing Network State TLV and all Node State TLVs.

NodeState-Req (Section 4.2.3): Provide requested data in a NetNode-Reply containing Node State and Node Data TLVs.

State-Reply (Section 4.2.4): If the message contains Node State TLVs that are more recent than local state (higher update number, different node data TLV hash, or we lack the node data altogether), and if the message also contains corresponding Node Data TLVs, update local state and reset Trickle. If the message is lacking Node Data TLVs for some Node State TLVs which are more recent than local state, reply with a NodeState-Req (Section 4.2.3) for the corresponding nodes.

Each node is responsible for publishing a valid set of data TLVs. When there is a change in a node's set of data TLVs, the update number MUST be incremented accordingly.

If a message containing Node State TLVs (Section 5.2.3) is received via unicast or multicast with the node's own node identifier and a higher update number than current local value, or the same update number and different hash, there is an error somewhere. A recommended default way to handle this is to attempt to assert local state by increasing the local update number to a value higher than that received and republish node data using the same node identifier. If this happens more than 3 times in 60 seconds and the local node

identifier is not globally unique, there may be more than one router with the same node identifier on the network. If HNCP security is not enabled, a new node identifier SHOULD be generated and node data republished accordingly. If HNCP security is enabled, this event is highly unlikely to occur as collision of identifier hashes for public keys is highly unlikely.

In all cases, if node data for any node changes, all Trickle instances MUST be considered inconsistent ($I = I_{min} + \text{timer reset}$).

4.4. Adding and Removing Neighbors

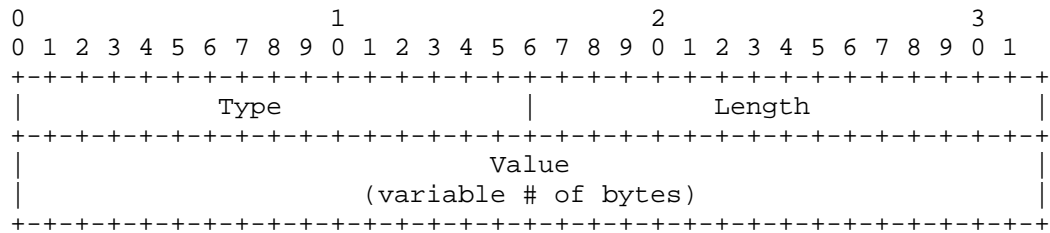
Whenever multicast message or unicast reply is received on a link from another node, the node should be added as Neighbor TLV (Section 5.2.6) for current node. If nothing (for example - no router advertisements, no HNCP traffic) is received from that neighbor in I_{max} seconds and the neighbor is not in neighbor discovery cache, and no layer 2 indication of presence is available, at least 3 attempts to ping it with request network state message (Section 4.2.2) SHOULD be sent with increasing timeouts (e.g. 1, 2, 4 seconds). If even after suitable period after the last message nothing is received, the Neighbor TLV MUST be removed so that there are no dangling neighbors. As an alternative, if there is a layer 2 unreachability notification of some sort available for either whole link or for individual neighbor, it MAY be used to immediately trigger removal of corresponding Neighbor TLV(s).

4.5. Purging Unreachable Nodes

When node data has changed, the neighbor graph SHOULD be traversed for each node following the bidirectional neighbor relationships. These are identified by looking for neighbor TLVs on both nodes, that have the remote node's identifier hash as $h(\text{neighbor node identifier})$, and local and neighbor link identifiers swapped. After the traverse, unreachable nodes SHOULD be purged after some grace period. During the grace period, the unreachable nodes MUST NOT be used for calculation of network state hash, or even be provided to any applications that need to use the whole TLV graph.

5. Type-Length-Value objects

Every TLV is encoded as 2 octet type, followed by 2 octet length (of the whole TLV, including header; 4 means no value), and then the value itself (if any). The actual length of TLV MUST be always divisible by 4; if the length of the value is not, zeroed padding bytes MUST be inserted at the end of TLV. The padding bytes MUST NOT be included in the length field.



Encoding of type=123 (0x7b) TLV with value 'x' (120 = 0x78): 007B
0005 7800 0000

Notation:

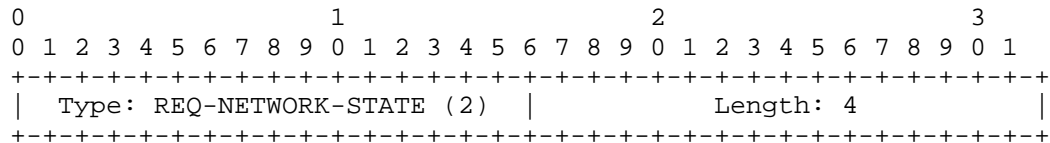
.. = octet string concatenation operation

H(x) = MD5 hash of x

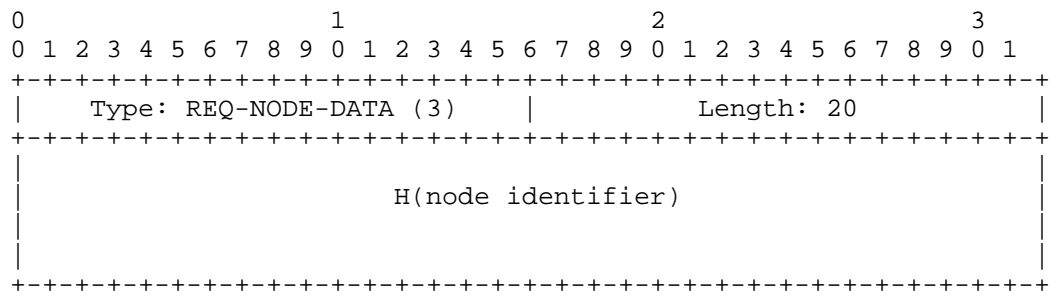
H-64(x) = H(x) truncated by taking just first 64 bits of the result.

5.1. Request TLVs (for use within unicast requests)

5.1.1. Request Network State TLV

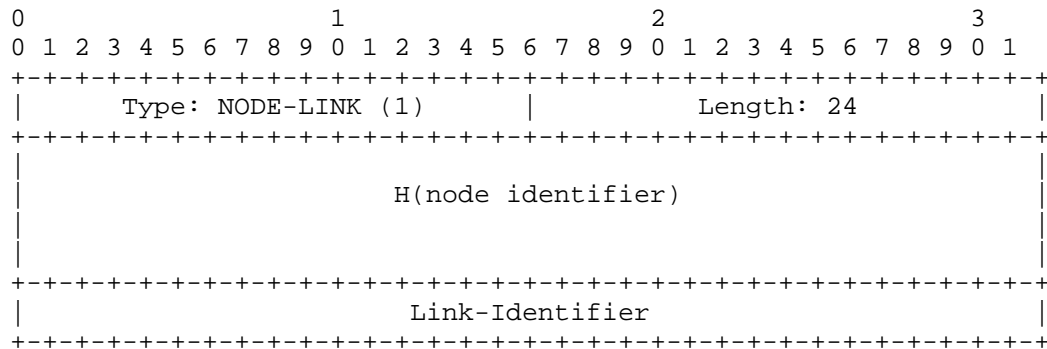


5.1.2. Request Node Data TLV

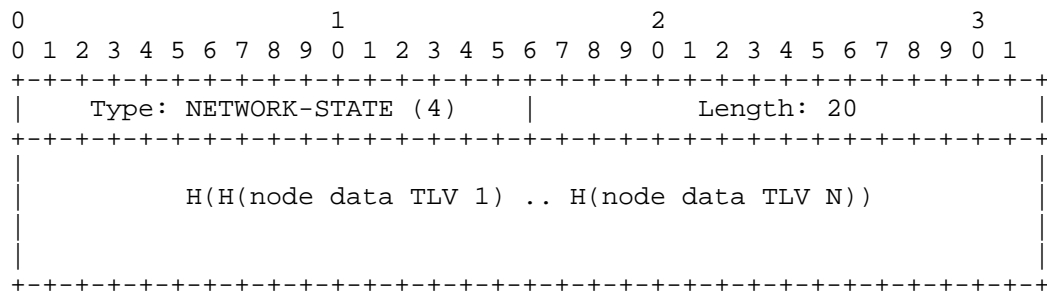


5.2. Data TLVs (for use in both multi- and unicast data)

5.2.1. Node Link TLV

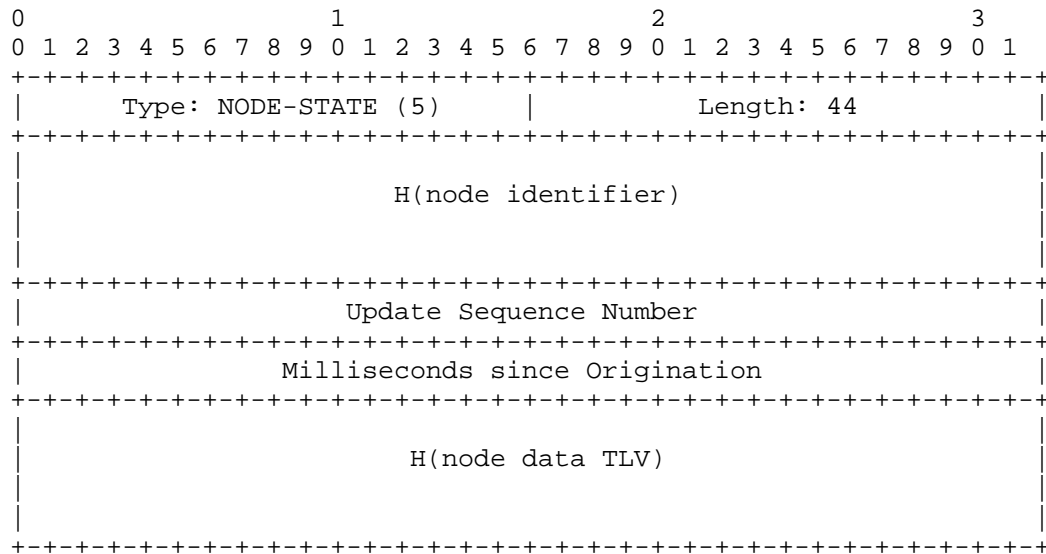


5.2.2. Network State TLV



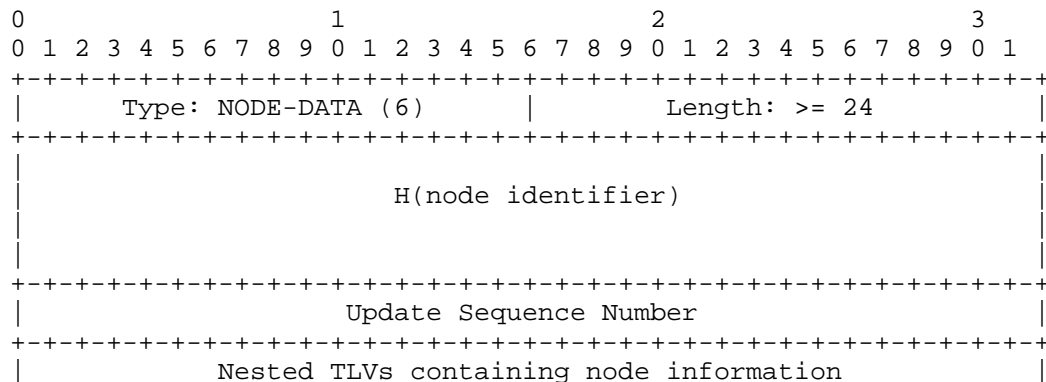
The Node Data TLVs are ordered for hashing by octet comparison of the corresponding node identifier hashes in ascending order.

5.2.3. Node State TLV



The whole network should have roughly the same idea about the time since origination, i.e. even the originating router should increment the time whenever it needs to send a new Node State TLV regarding itself without changing the corresponding Node Data TLV. This age value is not included within the Node Data TLV, however, as that is immutable and potentially signed by the originating node at the time of origination.

5.2.4. Node Data TLV



The Node Public Key TLV (Section 5.2.5) SHOULD be always included if signatures are ever used.

If signatures are in use, the Node Data TLV SHOULD also contain the originator's own Signature TLV (Section 5.5.2).

5.2.5. Node Public Key TLV (within Node Data TLV)

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|      Type: PUBLIC-KEY (7)      |      Length: >= 4      |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                               Public Key (raw node identifier)                               |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Public key data for the node. Only relevant if signatures are used. Can be used to verify that $H(\text{node identifier})$ in the received data for the node equals $H(\text{public key})$, and that the Signature TLVs are signed by appropriate public keys.

5.2.6. Neighbor TLV (within Node Data TLV)

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|      Type: NEIGHBOR (8)      |      Length: 28      |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                               H(neighbor node identifier)                               |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                               Neighbor Link Identifier                               |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                               Local Link Identifier                               |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

This TLV indicates that the node in question vouches that the specified neighbor is reachable by it on the local link id given. This reachability may be unidirectional (if no unicast exchanges have been performed with the neighbor). The presence of this TLV at least guarantees that the node publishing it has received traffic from the neighbor recently. For guaranteed bidirectional reachability, existence of both nodes' matching Neighbor TLVs should be checked.

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


```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   Type: CUSTOM-DATA (9)   |   Length: >= 12   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               H-64(URI)          |
|                               |                  |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               Opaque Data        |
|                               |                  |

```

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-64('http://example.com/author/json-for-hncp') .. '{"cool": "json
extension!"}'
```

or

```
V=H-64('mailto:author@example.com') .. '{"cool": "json extension!"}'
```

5.4. Version TLV (within Node Data TLV)

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   Type: VERSION (10)   |   Length: >= 8   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               Version          |
|                               |                  |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               User-agent        |
|                               |                  |

```

This TLV indicates which version of HNCP TLV binary structures is in use by this particular node. All TLVs within node data from nodes that do not publish version TLV, or with different Version value than locally supported one MUST be ignored (but forwarded). The user-agent is an optional human-readable UTF-8 string that can describe e.g. current hnetd version. This draft describes Version=1 TLVs.

5.5. Authentication TLVs

5.5.1. Certificate-related TLVs

TBD; should be probably some sort of certificate ID to be used in a lookup at most, as raw certificates will overflow easily IPv6 minimum MTU.

5.5.2. Signature TLV

TLV with T=0xFFFF, V=(TBD) public key algorithm based signature of all TLVs within current scope as well as the parent TLV header, if any. The assumed signature key is private key matching the public key of the the originator of node link TLV (if signature TLV is within main body of message), or that of the originator of the node data TLV (if signature TLV is within Node Data TLV)..

Given the ordering of TLVs, this TLV should be last one processed within current scope.

6. Border Discovery and Prefix Assignment

Using Default Border Definition [I-D.kline-homenet-default-perimeter] as a basis, this section defines border discovery algorithm specifics derived from the edge router interactions described in the Basic Requirements for IPv6 Customer Edge Routers [RFC7084]. The algorithm is designed to work for both IPv4 and IPv6 (single or dual-stack).

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

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

A homenet 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 set for an interface, it MUST be used.
2. Any of the following conditions indicate an interface MUST be considered external:
 1. A delegated prefix could be acquired by running a DHCPv6-client on the interface.
 2. An IPv4-address could be acquired by running a DHCP-client on the interface.

3. HNCP security is enabled and there are routers on the interface which could not be authenticated.
3. As default fallback, interface MUST be considered internal.

A router SHOULD 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 it MAY offer further categories which modify the local router behavior, such as:

Guest category: This is a specialization of the internal category which declares an interface used for untrusted clients. The router MUST NOT send or accept HNCP messages on these interfaces. Clients connected to these interfaces MUST NOT be able to reach devices inside the home network by default and instead SHOULD only be able to reach the internet.

Ad-hoc category: This is a specialization of the internal category which declares an interface to be in ad-hoc mode. This indicates to HNCP applications such as prefix assignment that links on this interface are potentially non-transitive.

Hybrid category: This is a specialization of the internal category in which the router still accepts external connections but does not do border discovery. It is assumed that the link is under control of a legacy, trustworthy non-HNCP router, still within the same home network. Detection of this category automatically is out of scope for this document, and therefore it MAY be supported only via manual configuration on a per-router basis.

A homenet router SHOULD provide basic connectivity to legacy CERS [RFC7084] connected to internal interfaces in order to allow coexistence with existing devices.

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 DHCP 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 possible default) in 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. Once a router detects an interface to be external it MUST stop any previously enabled internal forwarding. In addition it SHOULD announce the acquired information for use in the homenet as

described in later sections of this draft if the interface appears to be connected to an external network.

To distribute an external connection in the homenet an edge router announces one or more delegated prefixes and associated DHCP(v6)-encoded auxiliary information like recursive DNS-servers. Each external connection is announced using one container-TLV as follows:

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
| Type: EXTERNAL-CONNECTION (41) | Length: > 4 |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                               Nested TLVs                               |

```

Auxiliary connectivity information is encoded as a stream of DHCPv6-attributes or DHCP-attributes placed inside a TLV of type EXTERNAL-CONNECTION or DELEGATED-PREFIX (for IPv6 prefix-specific information). There MUST NOT be more than one instance of this TLV inside a container and the order of the DHCP(v6)-attributes contained within it MUST be preserved as long as the information contained does not change. The TLVs are encoded as follows:

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
| Type: DHCPV6-DATA (45) | Length: > 4 |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                               DHCPv6 attribute stream                               |

```

and

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
| Type: DHCP-DATA (44) | Length: > 4 |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                               DHCP attribute stream                               |

```

Each delegated prefix is encoded using one TLV inside an EXTERNAL-CONNECTION TLV. For external IPv4 connections the prefix is encoded in the form of an IPv4-mapped address [RFC4291] and is usually from a private address range [RFC1918]. The related TLV is defined as follows.

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|  Type: DELEGATED-PREFIX (42)  |           Length: >= 13           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               Valid until (milliseconds)           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               Preferred until (milliseconds)        |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Prefix Length |                                                     |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               Prefix Address [+ nested TLVs]        |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               |

```

Valid until is the time in milliseconds the delegated prefix is valid. The value is relative to the point in time the TLV is first announced.

Preferred until is the time in milliseconds the delegated prefix is preferred. The value is relative to the point in time the TLV is first announced.

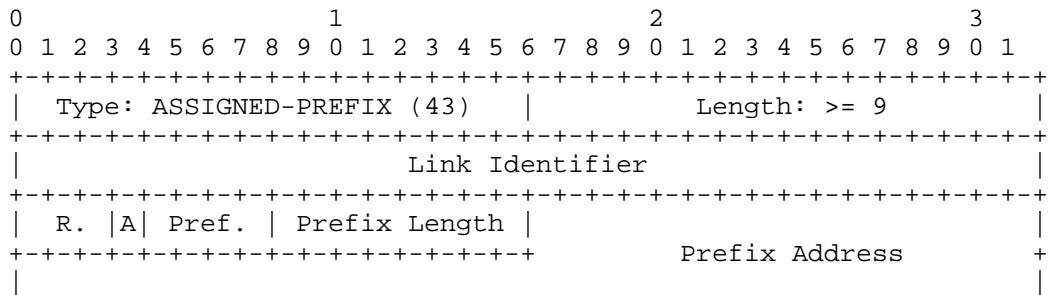
Prefix length specifies the number of significant bits in the prefix.

Prefix address is of variable length and contains the significant bits of the prefix padded with zeroes up to the next byte boundary.

Nested TLVs might contain prefix-specific information like DHCPv6-options.

In order for routers to use the distributed information, prefixes and addresses have to be assigned to the interior links of the homenet. A router MUST therefore implement the algorithm defined in Prefix and Address Assignment in a Home Network [I-D.pfister-homenet-prefix-assignment]. In order to announce the assigned prefixes the following TLVs are defined.

Each assigned prefix is given to an interior link and is encoded using one TLVs. Assigned IPv4 prefixes are stored as mapped IPv4-addresses. The TLV is defined as follows:



Link Identifier is the local HNCP identifier of the link the prefix is assigned to.

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

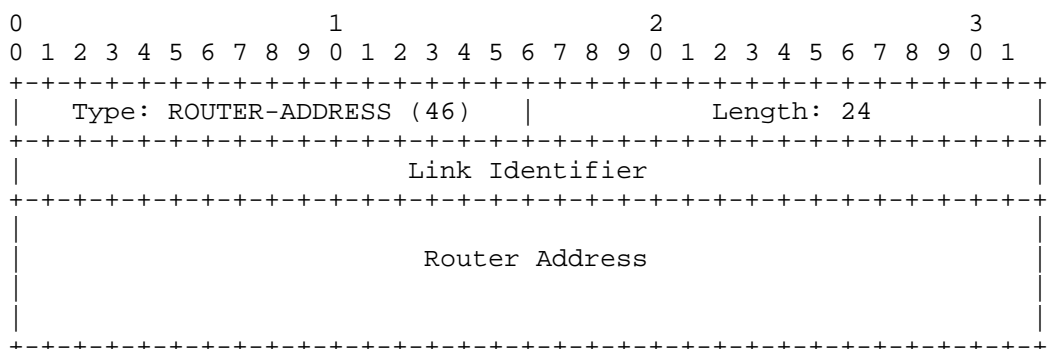
A is the authoritative flag which indicates that an assignment is enforced and ignores usual collision detection rules.

Pref. describes the preference of the assignment and can be used to differentiate the importance of a given assignment over others.

Prefix length specifies the number of significant bits in the prefix.

Prefix address is of variable length and contains the significant bits of the prefix padded with zeroes up to the next byte boundary.

In some cases (e.g. IPv4) the set of addresses is very limited and stateless mechanisms are not really suitable for address assignment. Therefore HNCP can manage router address in these cases by itself. Each router assigning an address to one of its interfaces announces one TLV of the following kind:



Link Identifier is the local HNCP identifier of the link the address is assigned to.

Router Address is the address assigned to one of the router interfaces.

7. DNS-based Service Discovery

Service discovery is generally limited to a local link. [I-D.stenberg-homenet-dnssd-hybrid-proxy-zeroconf] defines a mechanism to automatically extended DNS-based service discovery across multiple links within the home automatically. Following TLVs MAY be used to provide transport for that specification.

7.1. DNS Delegated Zone TLV

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
| Type: DNS-DELEGATED-ZONE (50) | Length: >= 21 |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Address                                     |
|                                     |                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Reserved |S|B|
+-----+-----+-----+-----+ Zone (DNS label sequence - variable length)
|

```

7.2. Domain Name TLV

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
| Type: DOMAIN-NAME (51) | Length: >= 4 |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Domain (DNS label sequence - variable length) |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

7.3. Router Name TLV

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   Type: ROUTER-NAME (52)   |           Length: >= 4           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           Name (not null-terminated - variable length)         |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

8. Routing support

8.1. Protocol Requirements

In order to be advertised for use within the Homenet, a routing protocol MUST:

Comply with Requirements and Use Cases for Source/Destination Routing [I-D.baker-rtgwg-src-dst-routing-use-cases].

Be configured with suitable defaults or have an auto-configuration mechanism (e.g. [I-D.acee-ospf-ospfv3-autoconfig]) such that it will run in a Homenet without requiring specific configuration from the Home user.

A router MUST NOT announce that it supports a certain routing protocol if its implementation of the routing protocol does not meet these requirements, e.g. it does not implement extensions that are necessary for compliance.

8.2. Announcement

Each router SHOULD announce all routing protocols that it is capable of supporting in the Homenet. It SHOULD assign a preference value for each protocol that indicates its desire to use said protocol over other protocols it supports and SHOULD make these values configurable.

Each router includes one HNCP TLV of type ROUTING-PROTOCOL for every such routing protocol. This TLV is defined as follows:

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   Type: ROUTING-PROTOCOL (60)   |           Length: 6           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   Protocol ID   |   Preference   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Protocol ID is one of:

- 0 = reserved
- 1 = Babel (dual-stack)
- 2 = OSPFv3 (dual-stack)
- 3 = IS-IS (dual-stack)
- 4 = RIP (dual-stack)

Preference is a value from 0 to 255. If a router is neutral about a routing protocol it SHOULD use the value 128, otherwise a lower value indicating lower preference or a higher value indicating higher preference respectively.

8.3. Protocol Selection

When HNCP detects that a router has joined or left the Homenet it MUST examine all advertised routing protocols and preference values from all routers in the Homenet in order to find the one routing protocol which:

1. Is understood by all routers in the homenet
2. Has the highest preference value among all routers (calculated as sum of preference values)
3. Has the highest protocol ID among those with the highest preference

If the router protocol selection results in the need to change from one routing protocol to another on the homenet, the router MUST stop the previously running protocol, remove associated routes, and start the new protocol in a graceful manner. If there is no common routing protocol available among all Homenet routers, routers MUST utilize the Fallback Mechanism (Section 8.4).

8.4. Fallback Mechanism

In cases where there is no commonly supported routing protocol available the following fallback algorithm is run to setup routing and preserve interoperability among the homenet. While not intended to replace a routing protocol, this mechanism provides a valid - but not necessarily optimal - routing topology. This algorithm uses the node and neighbor state already synchronized by HNCP, and therefore does not require any additional protocol message exchange.

1. Interpret the neighbor information received via HNCP as a graph of connected routers.
 2. Use breadth-first traversal to determine the next-hop and hop-count in the path to each router in the homenet:
 1. Start the traversal with the immediate neighbors of the router running the algorithm.
 2. Always visit the immediate neighbors of a router in ascending order of their router ID.
 3. Never visit a router more often than once.
 3. For each delegated prefix P of any router R in the homenet: Create a default route via the next-hop for R acquired in #2. Each such route MUST be source-restricted to only apply to traffic with a source address within P and its metric MUST reflect the hop-count to R.
 4. For each assigned prefix A of a router R: Create a route to A via the next-hop for R acquired in #2. Each such route MUST NOT be source-restricted.
 5. For the first router R visited in the traversal announcing an IPv4-uplink: Create a default IPv4-route via the next-hop for R acquired in #2.
 6. For each assigned IPv4-prefix A of a router R: Create an IPv4-route to A via the next-hop for R acquired in #2.
9. Security Considerations

General security issues for Home Networks are discussed at length in [I-D.ietf-homenet-arch]. The protocols used to setup IP in home networks today have very little security enabled within the control protocol itself. For example, DHCP has defined [RFC3118] to authenticate DHCP 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. HNCP aims to make security as easy as possible for the implementer by including built-in capabilities for authentication of node data being exchanged as well as the protocol messages themselves, but it is ultimately up to the shipping system to take advantage of the protocol constructs defined.

HNCP is designed to integrate with trusted bootstrapping [I-D.behringer-homenet-trust-bootstrap] including the ability to authenticate messages between nodes. This authentication can be used to securely define a border as well as protect against malicious attacks and spoofing attempts from inside or outside the border.

HNCP itself sends messages as (possibly authenticated) clear text which is as secure, or insecure, as the security of the link below as discussed in [I-D.kline-homenet-default-perimeter]. When no unique public key is available, a hardware fingerprint or equivalent to identify routers must be available for use by HNCP.

As HNCP messages are sent over UDP/IP, IPsec may be used for confidentiality or additional message authentication. However, this requires manually keyed IPsec per-port granularity for port IANA-UDP-PORT UDP traffic. Also, a pre-shared key has to be utilized in this case given IKE cannot be used with multicast traffic.

If no router can be trusted and additional guarantees about source of node status updates is necessary, real public and private keys should be used to create signatures and verify them in HNCP on both on per-node data TLVs as well as across the entire HNCP message. In this mode, care must be taken in rate limiting verification of invalid packets, as otherwise denial of service may occur due to exhaustion of computation resources.

As a performance optimization, instead of providing signatures for actual node data and the protocol messages themselves, it is also possible to provide signatures just for protocol messages. While this means it is no longer possible to verify the original source of the node data itself, as long as the set of routers is trusted (i.e., no router in the set has itself been hacked to provide malicious node data) then one can assume the node data is trusted because the router is trusted and the data arrived in a protected protocol message.

10. IANA Considerations

IANA should set up a registry (policy TBD) for HNCP TLV types, with following initial contents:

0: Reserved (should not happen on wire)

1: Node link

2: Request network state

3: Request node data

4: Network state
5: Node state
6: Node data
7: Node public key
8: Neighbor
9: Custom
10: Version
41: External connection
42: Delegated prefix
43: Assigned prefix
44: DHCP-data
45: DHCPv6-data
46: Router-address
50: DNS Delegated Zone
51: Domain name
52: Node name
60: Routing protocol
65535: Signature

HNCP will also require allocation of a UDP port number IANA-UDP-PORT, as well as IPv6 link-local multicast address IANA-MULTICAST-ADDRESS.

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

[I-D.pfister-homenet-prefix-assignment]

Pfister, P., Paterson, B., and J. Arkko, "Prefix and Address Assignment in a Home Network", draft-pfister-homenet-prefix-assignment-01 (work in progress), May 2014.

[I-D.stenberg-homenet-dnssd-hybrid-proxy-zeroconf]

Stenberg, M., "Auto-Configuration of a Network of Hybrid Unicast/Multicast DNS-Based Service Discovery Proxy Nodes", draft-stenberg-homenet-dnssd-hybrid-proxy-zeroconf-01 (work in progress), June 2014.

11.2. 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.

[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-13 (work in progress), March 2014.

[I-D.troan-homenet-sadr]

Troan, O. and L. Colitti, "IPv6 Multihoming with Source Address Dependent Routing (SADR)", draft-troan-homenet-sadr-01 (work in progress), September 2013.

[I-D.behringer-homenet-trust-bootstrap]

Behringer, M., Pritikin, M., and S. Bjarnason, "Bootstrapping Trust on a Homenet", draft-behringer-homenet-trust-bootstrap-02 (work in progress), February 2014.

[I-D.baker-rtgwg-src-dst-routing-use-cases]

Baker, F., "Requirements and Use Cases for Source/Destination Routing", draft-baker-rtgwg-src-dst-routing-use-cases-00 (work in progress), August 2013.

[I-D.kline-homenet-default-perimeter]

Kline, E., "Default Border Definition", draft-kline-homenet-default-perimeter-00 (work in progress), March 2013.

[I-D.acee-ospf-ospfv3-autoconfig]

Lindem, A. and J. Arkko, "OSPFv3 Auto-Configuration", draft-acee-ospf-ospfv3-autoconfig-03 (work in progress), July 2012.

11.3. URIs

[2] <http://www.openwrt.org>

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

Appendix A. Some Outstanding Issues

Should we use MD5 hashes, or EUI-64 node identifier to identify nodes?

Is there a case for non-link-local unicast? Currently explicitly stating this is link-local only protocol.

Consider if using Trickle with $k=1$ really pays off, as we need to do reachability checks if layer 2 does not provide them periodically in any case. Using Trickle with $k=\text{inf}$ would remove the need for unicast reachability checks, but at cost of extra multicast traffic. On the other hand, $N*(N-1)/2$ unicast reachability checks when lot of routers share a link is not appealing either.

Should we use something else than MD5 as hash? It IS somewhat insecure; however signature stuff (TBD) should rely on it mainly for security in any case, and MD5 is used in a non-security role.

Valid and preferred are now 32 bit millisecond and you cannot even represent a month in them; is this enough? Or should we switch to 32 bit seconds (or 64 bit milliseconds)?

Appendix B. Some Obvious Questions and Answers

Q: Why not use TCP?

A: It does not address the node discovery problem. It also leads to $N*(N-1)/2$ connections when N nodes share a link, which is awkward.

Q: Why not multicast-only?

A: It would require defining application level fragmentation scheme. Hopefully the data amounts used will stay small so we just trust unicast UDP to handle 'big enough' packets to contain single node's TLV data. On some link layers unicast is also much more reliable than multicast, especially for large packets.

Q: Why so long IDs? Why real hash even in insecure mode?

A: Scalability of protocol is not really affected by using real (=cryptographic) hash function.

Q: Why trust IPv6 fragmentation in unicast case? Why not do L7 fragmentation?

A: Because it will be there for a while at least. And while PMTU et al may be problems on open internet, in a home network environment UDP fragmentation should NOT be broken in the foreseeable future.

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: Why not doing (performance thing X, Y or Z)?

A: This is designed mostly to be minimal (only timers Trickle ones; everything triggered by Trickle-driven messages or local state changes). However, feel free to suggest better (even more minimal) design which works.

Appendix C. Changelog

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 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 and Juliusz Chroboczek for their contributions to the draft.

Authors' Addresses

Markus Stenberg
Helsinki 00930
Finland

Email: markus.stenberg@iki.fi

Steven Barth

Email: cyrus@openwrt.org

Interdomain Routing Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 28, 2014

N. Leymann
C. Heidemann
Deutsche Telekom AG
M. Wesserman
Painless Security
X. Xue
D. Zhang
Huawei
June 27, 2014

GRE Notifications for Hybrid Access
draft-lhwxz-gre-notifications-hybrid-access-00

Abstract

This document specifies a set of GRE (Generic Routing Encapsulation) extensions which enable operators to construct residential networks that are able to access the provider service through more than one hybrid access networks simultaneously in order to satisfy the higher bandwidth requirements.

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].

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 December 11, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	3
2. Terminology	4
3. GRE Solution Overview	4
4. IP Address Assignment	7
4.1. IPv4 Address Assignment	7
4.2. IPv6 Address Assignment	7
5. GRE Solution Function	9
5.1. GRE Tunnels Setup and Management	9
5.2. Packet-Based Traffic Overflow	12
5.3. Backward Compatibility	13
5.4. Bypassing Traffic Statistic	14
5.5. LTE and DSL Path Difference Consideration	15
6. GRE Control Message Definition	15
6.1. GRE Setup Request Message	17
6.2. GRE Setup Accept Message	17
6.3. GRE Setup Deny Message	18
6.4. GRE Hello Message	18
6.5. GRE Tear Down Message	18
6.6. GRE Notify Message	19
7. GRE Control Message Attribute Definitions	20
7.1. Client Identification Name (CIN)	21
7.2. Session ID	21
7.3. Timestamp	22
7.4. Bypass Traffic Rate	22
7.5. Filter List Package	23
7.6. RTT Difference Threshold	24
7.7. Bypass Bandwidth Check Interval	25
7.8. Switching to DSL Tunnel	26
7.9. Overflowing to LTE Tunnel	26
7.10. Hello Interval	26
7.11. Hello Retry Times	27

7.12. Idle Timeout	27
7.13. Error Code	28
7.14. DSL Link Failure	28
7.15. LTE Link Failure	28
7.16. IPv6 Prefix Assigned to Terminal Host	29
7.17. Subscribed DSL Upstream BW	30
7.18. Subscribed DSL Downstream BW	30
7.19. Delay Difference Threshold Violation	31
7.20. Delay Difference Threshold Compliance	31
7.21. Filter list ACK	32
7.22. End AVP	33
8. GRE Tunnels State Machine	33
9. IANA Considerations	34
10. Security Considerations	34
11. Acknowledgements	35
12. Normative References	35
Authors' Addresses	35

1. Introduction

In order to provide higher bandwidth for residential subscribers, operators prefer to bond the LTE network with DSL network to transfer the subscriber traffics. Especially, in some certain places (e.g. the old cities downtown), the DSL network is already overloading, even it is extremely difficult to be updated and rebuilt because of construction . To satisfy this requirement, HYbrid Access(HYA) architecture is designed in [I-D.lhwxyz-hybrid-access-network-architecture]. A solution is required to fill the gaps for operators deploying HYA.

This document proposes a packet-based HYA solution, which achieves bonding the hybrid access networks via extended Generic Routing Encapsulation (GRE)[RFC2890] protocol. This document presents the GRE protocol extensions required for HYA, specifically, those for signalling to setup, bond and management these GRE tunnels, signalling for reorder and reassemble customer traffics.

This remainder of this document is organized as follows. Section 2 lists the key terms used in this document. Section 3 outlines the overview of GRE solutions. In section 4, IP address assignment in HYA is described. Section 5 discusses the GRE solution functions. The definition of GRE control messages needed in HYA are listed in Section 6. The attributions used in GRE solutions are listed in Section 7. In Section 8, GRE Tunnels State MachineSection 8 is discussed.

2. Terminology

Customer Premise Equipment (CPE): A device that connects multiple hosts to provide connectivity to the service providers network.

DSL GRE Tunnel: The GRE tunnel between CPE DSL WAN and HAAP. The DSL GRE tunnel termination IP addresses are IP address of CPE DSL WAN interface and HAAP address.

Hybrid Access (HYA): Hybrid Access (HYA) is the bundling of two or more access lines over different technologies (e.g. DSL and LTE) to one Internet connection for end customers.

Hybrid Access Aggregation Point (HAAP): The HAAP which acts as a service termination and a service creation implements bonding mechanism and sets up a high speed Internet dual stack IP connection with CPE on top of two or more hybrid access technologies. The packet reorder, reassemble functions in packet-based solutions should be supported on HAAP.

HA Tunnel: HA Tunnel represents LTE GRE tunnel and DSL GRE tunnel defined between CPE and HAAP.

LTE GRE Tunnel: The GRE tunnel between CPE LTE WAN and HAAP. The LTE GRE tunnel termination IP addresses are IP address of CPE LTE WAN interface and HAAP address.

3. GRE Solution Overview

The GRE solution is proposed as a candidate solution for HYA based on per-packet traffic distribution mechanism. Only a dedicated GRE tunnel is setup over either hybrid access network between CPE and Hybrid Access Aggregation Point (HAAP), DSL GRE tunnel and LTE GRE tunnel. Figure 1. Bonding these GRE tunnels is preformed on CPE and HAAP. In addition, the types of packet distribution rules over hybrid accesses are deployed on both CPE and HAAP according to kinds of criteria (e.g., DSL load, failures, service list, etc).

To achieve these performances, the possible communications between CPE and HAAP are needed to achieve GRE tunnel setup, bonding and management, while to deploy and control the consistent traffic distribution for efficiency use of network resources. In addition, packet reorder, reassemble and fragmentation issues should be settled based on this communication[I-D.lhwxz-hybrid-access-network-architecture].

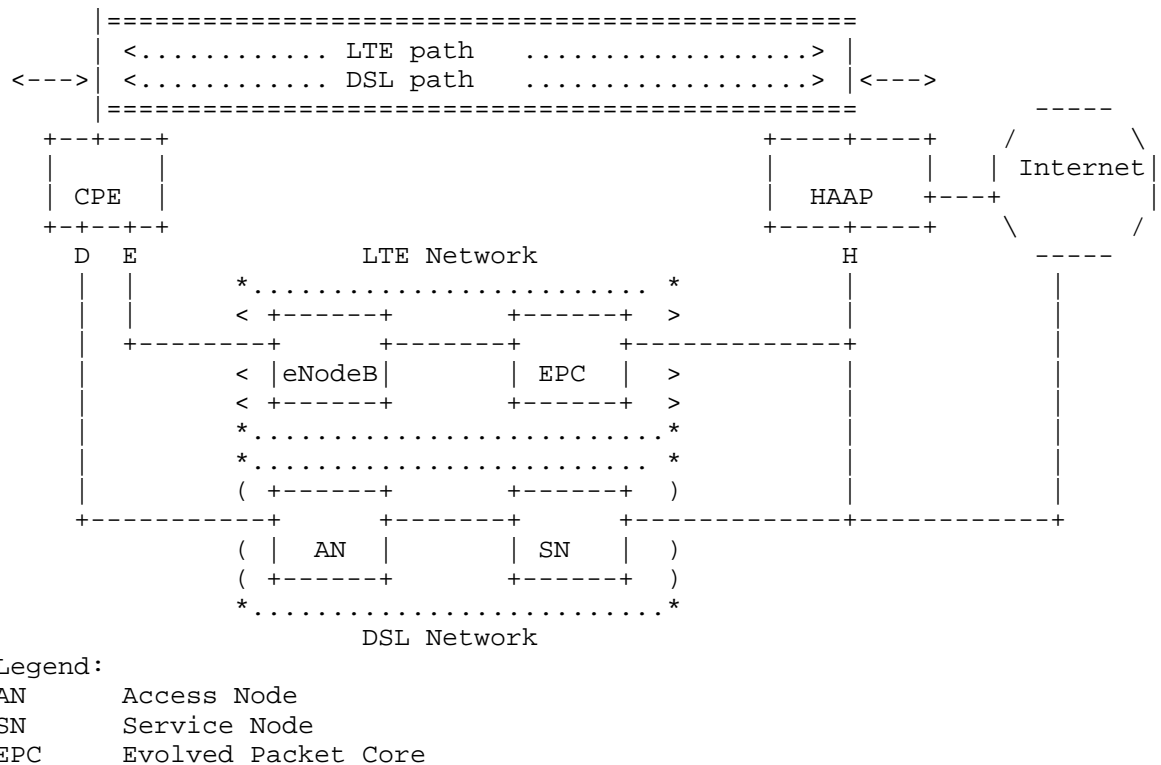
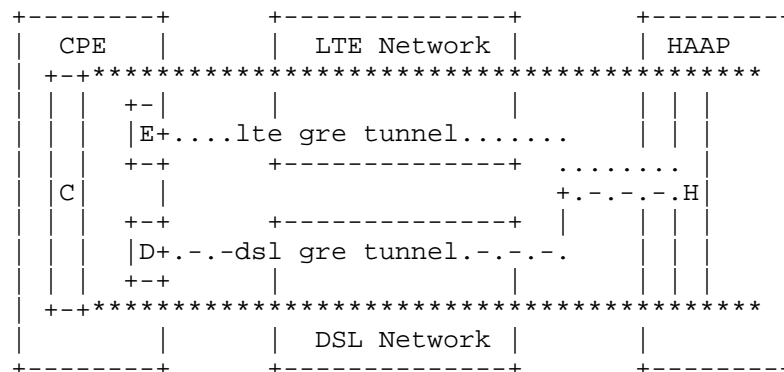


Figure 1: Hybrid Access Network Architecture

Once LTE and DSL GRE tunnels establishment and bonding procedure are completed, customer traffics can be distributed into LTE and/or DSL GRE tunnel based on traffic distribution rules on CPE and HAAP. The traffic encapsulation is shown in Figure 2.



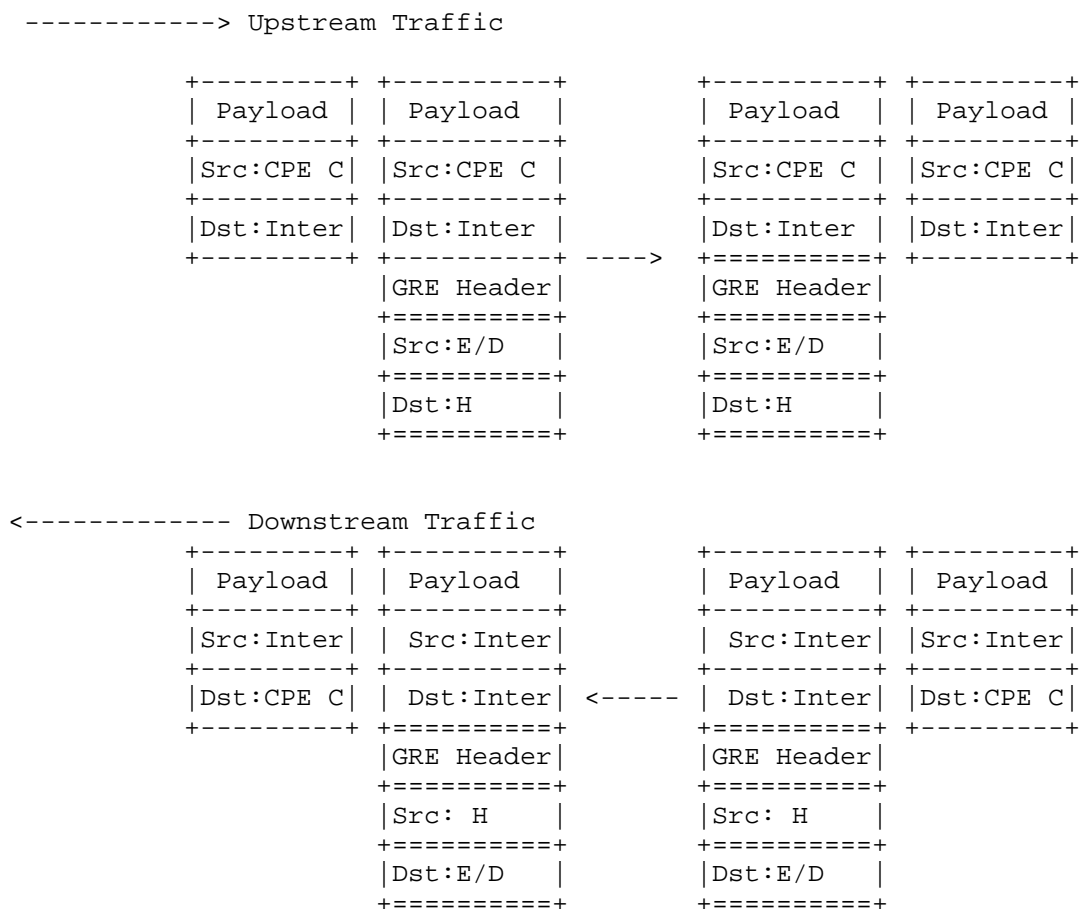


Figure 2: GRE Solution Overview

As shown in Figure 2 , particularly, the traffic going to upstream is encapsulated by GRE on CPE and decapsulated on HAAP. On the other side, HAAP encapsulates the downstream traffic by GRE which will be decapsulated on CPE. In order to clarify the details, the traffic forward actions is described following taking the upstream traffic as an example. A Internet service is initiated at CPE, whose source address is Src:CPE C, which is the public address of CPE assigned by HAAP, the destination address is dst: Inter, the specific Internet server address (e.g. google, youtube,etc). Receiving the upstream traffic, CPE encapsulates the packets of the upstream traffic by GRE tunnel, either LTE GRE tunnel header (Src: E and Dst: H) or DSL GRE tunnel header (Src:D and Dst:H) in order to balance the traffic between LTE and DSL network when DSL network is almost fully

occupied. When the GRE packets the HAAP, they will be decapsulated and then be forwarded as general IP packets.

4. IP Address Assignment

4.1. IPv4 Address Assignment

The IPv4 address assignment in Figure 2 are shown as follows:

- o E: CPE LTE WAN Interface IPv4 address (LTE GRE tunnel termination on CPE side)

In LTE network, during Packet Data Protocol (PDP) establishment[TS23.401], the PDN Gateway in LTE network will allocate IPv4 address to CPE LTE WAN interface , referred as E . This IPv4 address is used as LTE GRE tunnel termination's IPv4 address on CPE side.

- o D: CPE DSL WAN Interface IPv4 address (DSL GRE tunnel termination on CPE side)

In DSL network, during PPPoE exchanges [RFC2561], it is the DSL gateway (e.g. Broadband Network Gateway (BNG)) responsibility to allocate the IPv4 address to CPE DSL WAN interface. This IPv4 address is referred as D, which is used as DSL GRE tunnel termination's IPv4 address on CPE side.

- o C: CPE Public IPv4 address for route advertisement__

This address is assigned by HAAP acting as DHCPv4 server. CPE advertises this IPv4 address during Interior Gateway Protocol (IGP) exchanges for following service transmit. This is the IPv4 address used for the Internet communication.

- o H: HAAP IPv4 address (LTE/DSL GRE tunnel termination on HAAP side)

This address can be pre-configured statically on HAAP.

4.2. IPv6 Address Assignment

The IPv6 addresses in Figure 2 are shown as follows:

- o E: CPE LTE WAN Interface IPv6 prefix (LTE GRE tunnel termination on CPE side)

In LTE network the CPE LTE WAN interface gets assigned a specific IPv6 prefix (e.g. /64 prefix) by establishing PDP context with PGW, referred as D in Figure 2.

- o D: CPE DSL WAN Interface IPv6 prefix (DSL GRE tunnel termination on CPE side)

For IPv6 communication, the CPE DSL WAN interface is assigned a specific IPv6 prefix (e.g. /64 prefix) by BNG during PPPoE procedure.

- o C: CPE IPv6 prefix

This IPv6 prefix is assigned by HAAP. This address is stored both on CPE and HAAP. In this case, HAAP will act as DHCPv6 service.

- o H: HAAP IPv6 prefix (LTE/DSL GRE tunnel termination on HAAP side)

This address can be pre-configured statically on HAAP.

There may be two routing for terminal host traffic via the same CPE DSL WAN interface, one route is for bypass traffic without arriving HAAP in Section 5.3, the other route is for HYA traffic with arriving HAAP. So there must be two IPv6 address advertisement for one host in Internet. To achieve this purpose, the IPv6 prefix translation is deployed.

There are two scenarios:

1 DSL GRE Tunnel UP and LTE GRE Tunnel UP

Terminal Host will get a IPv6 prefix D-LAN from D prefix via SLAAC [RFC4862]. This prefix is used for DSL bypass traffic route advertisement.

IPv6 translation happens on HAAP. On HAAP, the terminal host IPv6 prefix D-LAN will be mapped to C, which is CPE IPv6 prefix assigned by HAAP. The C is used for HYA traffic route advertisement.

2 DSL GRE Tunnel Down and LTE GRE Tunnel UP

Terminal Host will get a IPv6 prefix C-LAN from C prefix via SLAAC [RFC4862]. This prefix is used for DSL bypass traffic route advertisement.

IPv6 translation happens on HAAP. On HAAP, the terminal host IPv6 prefix C-LAN will be mapped to C, which is CPE IPv6 prefix assigned by HAAP. The C is used for HYA traffic route advertisement.

5. GRE Solution Function

5.1. GRE Tunnels Setup and Management

In this document, the LTE and DSL GRE tunnels described in Figure 2 are established by GRE control messages exchanges between CPE and HAAP. The general procedures for the tunnels establishment are illustrated in the following diagram Figure 3.

The annotated ladder diagram shows CPE on the left, HAAP on the right. LTE and DSL network support customer traffic transmission as shown in the middle.

```

=====          ::::::::::          =====
      CPE          LTE/DSL          HAAP
=====          ::::::::::          =====

[...CPE obtains LTE WAN IF address during PDP from PGW....]
[...CPE obtains DSL WAN IF address during PPPoE from BNG...]
[..... CPE obtains HAAP address H via DNS .....]

[..... begin tunnel establishment and bond.....]

(..... begin lte gre tunnel establishment.....)

---- GRE Setup Request Message over LTE ---->
  ** Authentication and Authorization Passed **
<-- (1) GRE Setup Accept Message over LTE-----
      (carrying session ID)
  ** Authentication and Authorization Failed **
<--(2) GRE Setup Deny Message over LTE -----
if (1)
(..... lte gre tunnel establishment finishes ..... )
if (2)
(----- end -----)

---- Request CPE IP Address(C) (DHCP over LTE GRE) ---->
<--IP Address (C) Assigned to CPE(DHCP over LTE GRE)-----

(..... begin dsl gre tunnel establishment ..... )

---- GRE Setup Request Message over DSL ---->
(same session ID acquired during LTE establishment )
  ** Authentication and Authorization Passed **
<----(3) GRE Setup Accept Message over DSL ----
  ** Authentication and Authorization Failed **
<----(4) GRE Setup Deny Message over DSL -----
If (3)
(..... dsl gre tunnel establishment finishes.....)
(.....finish tunnel establishment and bond ..... )
if (4)
(----- end -----)

```

Figure 3: GRE Tunnel Establishment Procedure

The procedure of tunnel establishment is achieved by GRE control message exchanging. Meanwhile, the LTE and DSL GRE tunnels are bonded via the same "session ID" exchanged during the tunnel establishment procedure.

The details procedures are shown as follows:

1. CPE already gets DSL WAN interface IP address through PPPoE from BRAS and LTE WAN interface IP address through PDP from PGW.
2. CPE request DNS resolution for HAAP domain name via DSL WAN or LTE WAN interface, DNS server will return a corresponding HAAP IP address which can be pre-configured by operators.
3. Then CPE tries to setup the tunnels and bundling them. CPE will setup LTE GRE tunnel before DSL GRE tunnel. CPE sends GRE Tunnel Setup Request message to HAAP via LTE WAN interface.
4. The HAAP receives the message and then initiates the Authentication and Authorization procedure in order to check whether CPE is trusted for PGW. It is similar like UE authentication in [TS23.401].
5. After authentication and authorization succeed, HAAP then replies GRE Tunnel Setup Accept message to CPE via LTE. Specially, Session ID generated randomly by HAAP will be carried in this message, which is used for bonding LTE GRE tunnel and DSL GRE tunnel for one subscriber later. If authentication and authorization failed, HAAP must send the GRE Setup Deny message to CPE over LTE, the tunnel establishment procedure must be tore down.
6. After LTE GRE tunnel setup is success, CPE begins to obtain C address defined in Section 4 from HAAP through DHCP over LTE GRE tunnel. At the same time, CPE begins to setup DSL GRE tunnel.
7. CPE sends GRE Setup Request message with HAAP address as the destination IP of GRE via DSL WAN interface, carrying the same session ID received from HAAP in Step 5.
8. The HAAP receives the message and then initiates the Authentication and Authorization procedure in order to check whether CPE is trusted for BRAS and validate the HYA service rights for CPE.
9. After authentication and authorization succeed, HAAP sends GRE Setup Accept message to CPE via DSL. CPE then bundle the two GRE tunnels based on same Session ID.
10. CPE sends GRE Notify message via DSL WAN immediately after the DSL GRE tunnel setup successfully in order to inform the DSL bypass bandwidth to HAAP. More details is shown in Section 6.

For management and control motivations, GRE tunnel management process message exchanges between CPE and HAAP are needed, shown in the following figureFigure 4.

```

=====          ::::::::::          =====
      CPE          LTE/DSL          HAAP
=====          ::::::::::          =====

(..... lte/dsl tunnel failure detection and keepalive...)
  ----- GRE Hello Message over LTE ----->
<----- GRE Hello Message over LTE -----
  ----- GRE Hello Message over DSL ----->
<----- GRE Hello Message over DSL -----

(.....lte/dsl tunnel information inform.....)
  ----- GRE Notify Message over LTE----->
<----- GRE Notify Message over LTE -----
  ----- GRE Notify Message over DSL----->
<----- GRE Notify Message over DSL -----

( ..... lte/dsl tunnel teardown ..... )
<----- GRE Tear Down Message over LTE -----
<----- GRE Tear Down Message over DSL -----

```

Figure 4: GRE Tunnel Management Procedure

GRE Hello messages exchange between CPE and HAAP for LTE/DSL tunnel failure detection and keep-alive. GRE Notify message is used to inform status/information (e.g., dsl network status, service list for HYA, etc) between CPE and HAAP. A notify acknowledgement (ACK) via GRE Notify message and retransmission mechanism can be used to provide certain level reliable transport capability. For maintenance reasons, GRE Tear Down message can be used by HAAP to terminate the bond LTE GRE tunnel and DSL GRE tunnel for some reasons because of network failures. The detailed control messages are proposed in Section 6.1.

5.2. Packet-Based Traffic Overflow

In this document, traffic distribution between the established and bond LTE and DSL GRE tunnel is packet-based overflow. The packet-based traffic overflow mechanism includes two requirements, cheapest path used first (e.g., DSL GRE tunnel Figure 2) and traffics overflowed when cheapest path is almost fully occupied. To satisfy these requirements, Two Rate Three Color Marker (trTCM) [RFC2698] can be used.

Two token buckets based on DSL and LTE resource are used to meter if the packets is overflowed or not. The details rate configuration is based on the operators' requirement, which is out of the scope. Clearly, the packet can be marked with yellow if the packet is overflowed, otherwise the packet is marked with green based on [RFC2698]. Then the colored based policy routing is executed on CPE and HAAP. The packet will be routed into the corresponding tunnel based on the marked color. For example, yellow color packet will be routed to LTE GRE tunnel; green color packet will be routed into DSL GRE tunnel. The GRE IP header is used to encapsulate the traffic on CPE and HAAP as shown in . (Figure 2).

On the received side, the packets encapsulated in GRE will come from DSL GRE tunnel and/or LTE GRE tunnel. Due to different transporting delivery caused by LTE and DSL paths, the packets in the same flow may reach out of the order. Consequentially the packets will be sent to a buffer for reordering based on the sequence information in GRE header, details in Section 6. After reordering, the GRE header will be removed and the packet will be sent to the ordinary IP packet processing.

5.3. Backward Compatibility

The solution should satisfy the backward compatibility requirements. While deploying HYA architecture, the existing services must not be influenced. For example, IPTV traffic must be remained into the DSL path only for performance reasons, instead of LTE tunnel. In addition some control messages (e.g. for TR069/ACS, DNS etc.) might not be reachable through the HAAP as well due to control and management entities deployment scenario in the network. These kinds of services can be defined and managed by operators during HYA deployment.

In this document, the mechanism must be defined for deploying and maintaining the list of these kinds of traffic. The negotiation between HG and HAAP Figure 5 is described for this purpose.

During network arrangement, operators may configure this service list. HAAP provision the information to CPE via LTE/DSL GRE tunnel. And the list must be updateable during the established tunnel. At each time when CPE try to establish the tunnel, the list is pushed by HAAP. CPE will flash the the list if it have a previous one. If the list is taken some errors during list flashing, CPE should keep the previous one and reply the error code to HAAP via GRE Notify message. The errors include download unsuccessfully, incorrect format, wrong syntax etc defined in Section 6.

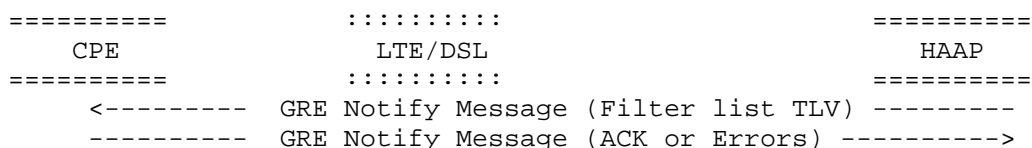


Figure 5: GRE Tunnel Service List Management

As shown Figure 5, only one GRE tunnel (LTE/GRE) will be used for one time transmission of GRE Notify message carrying the service list, and each notification will be replied by a notification ACK. If several times of transmission failure for notification, the tunnel for sending notification will be switched to the other one.

HG will validate the received filter list packet, if no error found, CPE will reply GRE Notify message as ACK to HAAP. So HAAP directly stops to send the following filter list packet, that means this time of filter list notification is completed successfully. If any error found, CPE will reply GRE Notify message as errors feedback, HAAP will try to send it again or stop it. The details are described in Section 6.

In case of large size of the service list, multiple GRE Notify messages to CPE are needed to carry multiple fragments of the list. Each of these GRE Notify message needs a notification ACK.

5.4. Bypassing Traffic Statistic

Bypassing Traffic means that the traffic MUST bypass the HYA GRE tunnel but directly over DSL WAN interface as mentioned filter list in Section 5.3, and this happens on the CPE. The traffic bypass behavior is accomplished by implementing a routing table on CPE. Distinctly, part of DSL bandwidth is already occupied by these types of bypass traffic with higher priority. As a result, only the DSL bandwidth left can be used for HYA DSL GRE tunnel.

The solution must consider how to meter the bypassing traffic statistic on DSL bandwidth and adjust the free resource left in DSL for HYA. The DSL bandwidth for HYA must be adjusted dynamically when bypass traffics are presenting. CPE can check the bypass traffic rate periodically, and notify the parameters to the HAAP. HAAP can adjust the token buckets for packet overflow action later on defined in Section 5.2.

5.5. LTE and DSL Path Difference Consideration

In HYA, LTE and DSL tunnel may have different characters, such as rates, delay and MTU which cause the throughput and traffic fragmentation issues. These differences should be considered during the GRE solution design.

The rate, Round Trip Time (RTT) /delay of a DSL link is relatively fixed, but the RTT/delay character of an LTE link vary over time. When the DSL and LTE link are combined in HYA, the CPE has a larger combined bandwidth (DSL_BW + LTE_BW), but the RTT/delay of the bonded tunnels may become bigger for customer traffics. The maximum RTT/delay of customer HYA traffic is equal to bigger one of the LTE and DSL links. Usually, the buffering size for packet reorder is related to the RTT/delay difference between both LTE and DSL link. If the RTT/delay difference is too big, the buffer size will be too huge to be achieved on CPE/HAAP. In this case, the bandwidth efficiency of the HYA will disappeared comparing the bigger RTT/delay and huge buffer requirement.

The MTU difference may impact the packet fragmentation and reorder. The minimum MTU on DSL path is PPPoE MTU, which is 1492. The minimum MTU on LTE path is UGW MTU, which is 1436. In HYA, the maximum tunnel MTU is LTE MTU minus GRE overhead. Static calculation for GRE tunnel MTU sized based on DSL path MTU and LTE path MTU is configured. MSS adjustment for TCP on CPE based on the calculation in order to avoid IP fragmentation on both GRE outer IP layer and inner IP layer.

6. GRE Control Message Definition

In this section, GRE encapsulation control messages are defined for negotiation between CPE and HAAP for the LTE and DSL tunnel establishment, bond, management, etc, which are not standardized yet. The GRE control messages format are according to [RFC2890]. The GRE header as described in Figure 6 indicates a control protocol with the Protocol Type section set to 0x0101 in this document.

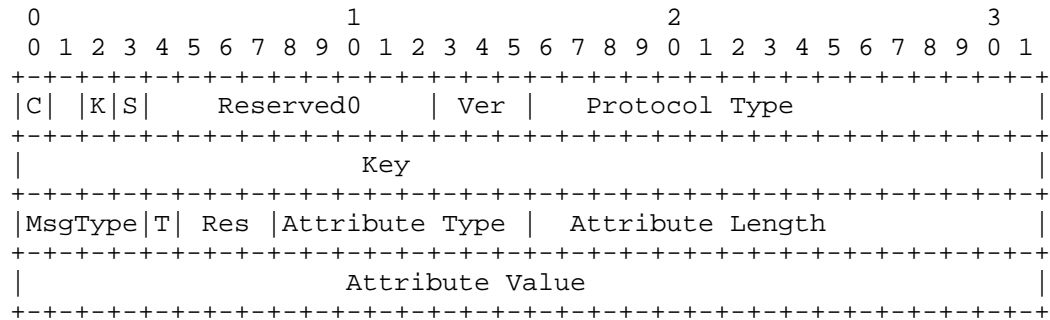


Figure 6: GRE Header Format

Protocol Type (2 octets)

The Protocol Type field identifies the GRE protocol for HYA network. The value 0x0101 is proposed.

Message Type (MesType) (4 bits)

The Message Type field identifies GRE protocol control messages for HYA network. Right now, there are 6 valid types of GRE control message mentioned , shown as belowFigure 7:

Control Message Family	Type
=====	=====
GRE Setup Request	1
GRE Setup Accept	2
GRE Setup Deny	3
GRE Hello	4
GRE Tear Down	5
GRE Notify	6
Reserved	0,7-15

Figure 7: GRE Control Messages

Tunnel Type (T) (1 bit)

If the Tunnel Type bit is set to 1, then it indicates that this control message is used for the DSL GRE tunnel. Otherwise it indicates that this control message is used for the LTE GRE tunnel.

Attribute Type (1 octet)

Attribute Type indicates the type of the appended attributes included in the GRE header. The types of attributes are defined in Section 7.

Attribute Length (2 octets)

Attribute Length field indicates the length of the attribute by byte.

Attribute Value (variable)

Attribute Value field includes the value of the attribute.

6.1. GRE Setup Request Message

GRE Setup Request message is sent by CPE to HAAP via LTE and DSL WAN in order to set up LTE and DSL GRE tunnel.

The following attributions MUST be included in the GRE Setup Request Message.

- o Client Identification Name (CIN) Figure 10. Only the GRE Tunnel Setup Request Message through LTE WAN must contain the CIN.
- o Session ID Figure 11. CPE must encapsulate the Session ID attribute in GRE Setup Request message via DSL WAN. This Session ID is generated by HAAP during LTE tunnel establishment Figure 3. The value in Session ID attribute must be same via both DSL and LTE WAN. In addition, when LTE GRE tunnel recovery from failure while DSL GRE tunnel exists, the re-established LTE tunnel request needs to carry the Session ID Attribute.
- o End AVP, see Section 7.

6.2. GRE Setup Accept Message

HAAP sends GRE Setup Accept Message to CPE if HAAP accepts associated GRE Setup Request from CPE. The routing path of a pair of GRE Setup Request message and GRE Setup Accept message must be the same, either LTE or DSL.

The following attributions MUST be included in the GRE Setup Accept Message via LTE WAN.

- o Session ID Figure 11, HAAP generates a session ID for a CPE and sends the Session ID attribute to CPE LTE WAN via GRE Setup Accept Message.
- o RTT Difference Threshold Attribute Figure 16, see Section 7.6.
- o Bypass Bandwidth Check Interval Figure 17, see Section 7.7.
- o Hello Interval Figure 18, see Section 7.10.

- o Hello Retry TimesFigure 19, see Section 7.11.
- o Idle TimeoutFigure 20, see Section 7.12.
- o Delay Difference Threshold Violation, see Section 7.19
- o Delay Difference Threshold Compliance, see Section 7.20
- o End AVP, see Section 7.22

The following attributions MUST be included in the GRE Setup Accept Message via DSL WAN.

- o Subscribed DSL Upstream BWFigure 23, see Section 7.17.
- o Subscribed DSL Downstream BWFigure 24, see Section 7.18.
- o End AVP, see Section 7.22

6.3. GRE Setup Deny Message

HAAP will send GRE Setup Deny Message to CPE through LTE and/or DSL path if HAAP denies the GRE Setup Request for LTE and/or DSL GRE tunnel from CPE.

The following attributions MUST be included in the GRE Setup Deny Message.

- o Error CodeFigure 21, see Section 7.13.
- o End AVP, see Section 7.22.

6.4. GRE Hello Message

The GRE Hello Message is used for CPE and HAAP on both LTE GRE tunnel and DSL GRE tunnel for failure detection and keepalive of the tunnel.

The following attributes MUST be included in the GRE Hello Message.

- o TimestampFigure 12, see Section 7.3.
- o End AVP, see Section 7.22.

6.5. GRE Tear Down Message

GRE Tear down message is used to maintain the state and can only be send from HAAP to CPE to terminate the established LTE and/or DSL tunnels.

The following attributes MUST be included in the GRE Tear Down Message.

- o Error CodeFigure 21, see Section 7.13.
- o End AVP, see Section 7.22.

6.6. GRE Notify Message

GRE notify message is used to inform status/information changing and the filter list information between CPE and HAAP.

The following attributes MUST be included in the GRE Notify Message via both LTE and DSL WAN.

- o End AVP, see Section 7.22.

The following attributes MAY be included in the GRE Notify Message via LTE WAN .

- o Filter list packageFigure 14, see Section 7.5.
- o DSL link failure, see Section 7.14.
- o IPv6 prefix assigned to terminal hostFigure 22, see Section 7.16.
- o Filter list ACKFigure 14, see Section 7.21.

The following attributes MAY be included in the GER Notify Message via DSL WAN.

- o Bypass traffic rateFigure 13, see Section 7.4.
- o Filter list packageFigure 14, see Section 7.5.
- o Switching to DSL tunnel, see Section 7.8.
- o Overflowing to LTE tunnel, see Section 7.9.
- o LTE link failure, see Section 7.15.
- o IPv6 prefix assigned to terminal hostFigure 22, see Section 7.16.
- o Filter list ACKFigure 14, see Section 7.21.

7. GRE Control Message Attribute Definitions

All the attributions are identified by the Type, Length, Value field, shown as below Figure 8. The 8-bits Type field identifies the type of the attribution.

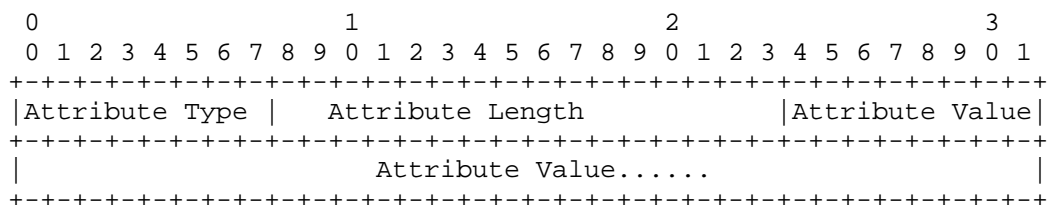


Figure 8: GRE Control Message Attribute Definitions

The following GRE control message attributes for HYA are defined in this document Figure 9 .

Control Message Family =====	Type =====
CIN	3
Session ID	4
Timestamp	5
Bypass Traffic Rate	6
Filter List Package	8
RTT Difference Threshold	9
Bypass Bandwidth Check Interval	10
Switching to DSL Tunnel	11
Overflowing to LTE Tunnel	12
Hello Interval	14
Hello Retry Times	15
Idle Timeout	16
Error Code	17
DSL Link Failure	18
LTE Link Failure	19
IPv6 Prefix Assigned to Terminal Host	21
Subscribed DSL Upstream BW	22
Subscribed DSL Downstream BW	23
Delay Difference Threshold Violation	24
Delay Difference Threshold Compliance	25
Filter list ACK	30
End AVP	255
Reserved	

Figure 9: GRE Control Message Attributes

7.1. Client Identification Name (CIN)

CIN is used to identified the RG in operator network. CIN is sent to HAAP by CPE for authentication and authorization purpose. It is similar like UE authentication in [TS23.401].Any CPE must transmit a CIN during the tunnel request procedure for authentication. CIN must be unique for each CPE in operator's network.

The attribute contains the following value Figure 10:

```

      0               1               2               3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               CIN                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Figure 10: CIN Attribute

Type:3 for CIN Attribute

Length: 40 Bytes

CIN: String defined by operators

7.2. Session ID

Session ID attribute is used to bind the DSL tunnel and LTE tunnel together for individual CPE. Session ID 32bit value is generated by HAAP, and unique within a HAAP. It is used to identify a certain subscriber CPE.

HAAP sends this attribute to requesting CPE LTE WAN via GRE Setup Accept message, then CPE encapsulates this attribute in GRE Setup Request through DSL WAN. With this information, CPE and HAAP can bind these two tunnels together. When LTE recovery from failure with DSL tunnel exists, the re-established LTE tunnel request needs to carry the Session ID.

The attribute contains the following value Figure 11:

```

      0               1               2               3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               Session ID                       |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Figure 11: Session ID Attribute

Type:4 for Session ID Attribute

Length: 4 Bytes

Session ID: String value generated by HAAP to identify a certain CPE.

7.3. Timestamp

The Timestamp attribute is used for Round-Trip Time (RTT) RTT calculation.

The attribute contains the following value Figure 12.

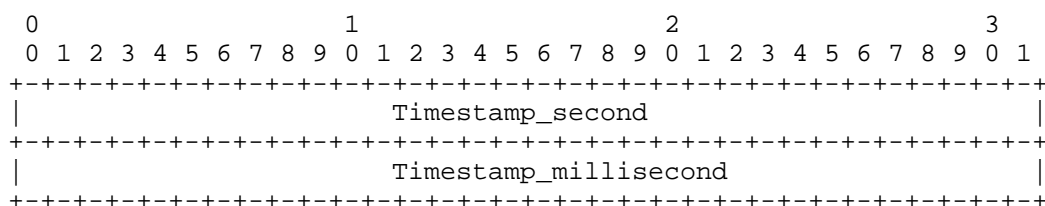


Figure 12: Timestamp Attribute

Type:5 for Timestamp Attribute

Length: 8 Bytes

Session ID: The higher order 4 bytes is seconds, the low-order 4 bytes is millisecond

7.4. Bypass Traffic Rate

The Bypass Traffic Rate attribute is used by HG to notify HAAP of the bypass traffic rate on CPE, such as IPTV, DNS, etc, see Section 5.4 for details. HAAP will calculate the available DSL bandwidth for HYA DSL GRE tunnel based on this information.

The attribute contains the following valuesFigure 13.

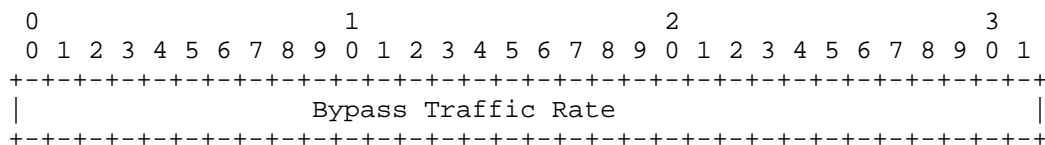


Figure 13: Bypass Traffic Rate Attribute

Type:6 for BypassTraffic Rate Attribute

Length: 4 Bytes

Bypass Traffic Rate: A 4-bytes length integer to identify the resource already occupied in DSL by kinds of bypass traffic referred as Section 5.4 .The CPE will check the bypass traffic rate periodically, if the bypass traffic rate difference is greater than specified percentage of the DSL bandwidth, and then notify the bypass traffic rate to the HAAP. HAAP can adjust the token buckets for packet overflow action later on Section 5.4.

7.5. Filter List Package

The Filter List Package is the collection of the services list which MUST not be routed through HYA, but directly over the specific interface mentioned in Section 5.3. The filter service list is configured on CPE by HAAP. This attribute is the collection of filter list TLVs, each TLV carries one kinds of filter service list.

The attribute contains the following values

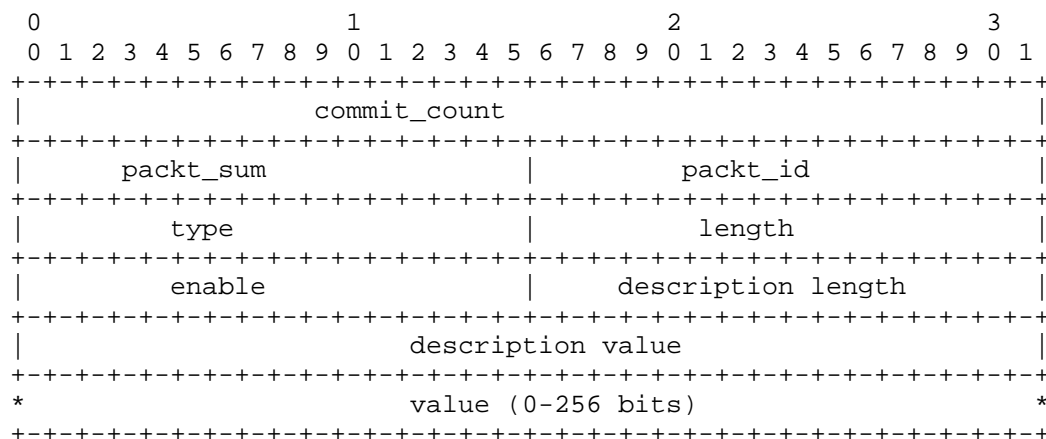


Figure 14: Filter List Package Attribute

Type: 8 for Filter List Package Attribute

Length: <= 969 Bytes

Commit_count: It is used to identify the Filter list version. If the Filter list recieved from HAAP changed, the commit_count will be updated. CPE will refresh the previous filter list.

Packet_sum: If the filter list packet is larger than the MTU and should be divided into multiple fragments, the Packet_sum indicate the fragments numbers of the filter list packet.

Packet_ID: The index of the multiple fragments.

Type: Several filter list type can be defined, which is described as following Figure 15.

Length: The length of the specific type of filter list.

Enable: Indicate this type of filter list is enabled. Only can be 1(Enabled) or 0 (Unenabled), other values are reserved.

Description Length: The length of this type of filter list description, the unit is byte.

Description Value: The value of this type of the filter list

Value: Value of the specific type of filter list.

Filter List	Type
=====	=====
Fully Qualified Domain Name	1
DSCP	2
Destination Port	3
Destination IP	4
Destination IP&Port	5
Source Port	6
Source IP	7
Source IP&Port	8
Source Mac	9
Protocol	10
Source IP Range	11
Destination IP Range	12
Source IP Range&Port	13
Destination IP Range&Port	14
Reserved	

Figure 15: Filter List Type

7.6. RTT Difference Threshold

The difference RTT/delay between DSL and LTE should impact the HYA network efficiency, mentioned in Section 5.5. So the acceptable RTT difference threshold in HYA must be defined. This value is signed to CPE by HAAP. When the RTT difference exceeds the configured RTT

difference threshold, CPE may changing the traffic distribution into DSL only rather than LTE GRE tunnel.

The attribute contains the following valuesFigure 16

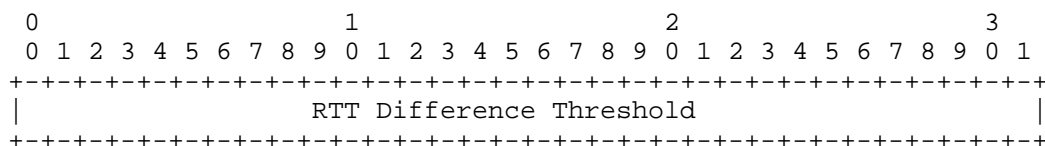


Figure 16: RTT Difference Threshold

Type: 9 for RTT Difference Threshold Attribute

Length: 4 Bytes

RTT Difference Threshold: The unit of this integer value is ms (milliseconds). This value is configurable, the value range can be from 0~1200ms, changing step is 1ms.

7.7. Bypass Bandwidth Check Interval

The Bypass Bandwidth Check Interval is assigned to CPE by HAAP. Based on requirement in Section 5.4, CPE will check the bypass bandwidth on DSL path after this interval.

The attribute contains the following valueFigure 17:

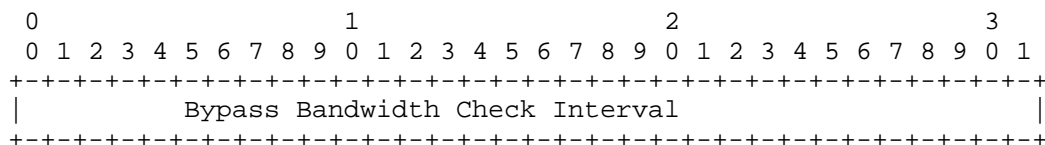


Figure 17: Bypass Bandwidth Check Interval Attribute

Type: 10 for Bypass Bandwidth Check Interval Attribute

Length: 4 Bytes

Bypass Bandwidth Check Interval: Integer as seconds. This value is configurable, the range is from 10-300s, changing step is 1s.

7.8. Switching to DSL Tunnel

The Switching to DSL Tunnel is used by CPE to notify HAAP to switch the traffic to DSL only. When the RTT difference between DSL and LTE tunnel exceeds the RTT difference thresholdFigure 16 3 times (default value), the CPE will send a Notify message with "Switching to DSL tunnel" attribute to HAAP. Then the traffic will be sent through the DSL tunnel only no matter HAAP or CPE.

There is no value in this attribute.

Type: 11 for Switching to DSL Tunnel

Length: 0

7.9. Overflowing to LTE Tunnel

The Overflowing to LTE Tunnel is used by CPE to notify HAAP to overflow the traffic to LTE tunnel. When the RTT difference between DSL and LTE tunnel is lower than the RTT difference thresholdFigure 16 3 times (default value), the CPE will send a a Notify message with "Overflowing to LTE tunnel" attribute to HAAP. Then the traffic can overflow to the LTE tunnel.

There is no value in this attribute.

Type: 12 for Overflowing to LTE Tunnel

Length: 0

7.10. Hello Interval

The Hello Interval is configured to CPE by HAAP. The GRE Hello message between CPE and HAAP will be negotiated in each hello interval period.

The attribute contains the following valueFigure 18:

```

      0               1               2               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Hello Interval                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Figure 18: Hello Interval Attribute

Type: 14 for Hello Interval Attribute

Length: 4 Bytes

Hello Interval: Integer. The unit of this value is second. This value should be configurable, with range 1~100s, changing step is 1s.

7.11. Hello Retry Times

The Hello Retry Times is configured to CPE by HAAP. The GRE Hello message between CPE and HAAP will be retried several times defined in this attribute.

The attribute contains the following valueFigure 19.

```

      0               1               2               3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Hello Retry Times                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Figure 19: Hello Retry Times Attribute

Type: 15 for Hello Retry Times Attribute

Length: 4 Bytes

Hello Retry Times: Integer. It is the times about the GRE Hello Message retry. This value is configurable, the value range is from 3~10, changing step is 1.

7.12. Idle Timeout

The Idle Timeout is configured on CPE by HAAP. If GRE tunnels are already established via DSL and LTE, idle timeoutFigure 28 will occur. All tunnels must be terminated if LTE/DSL tunnel isn't restored within a period time (e.g., idle timeout).

The attribute contains the following valueFigure 20.

```

      0               1               2               3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Idle Timeout                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Figure 20: Idle Timeout Attribute

Type: 16 for Idle Timeout Attribute

Length: 4 Bytes

Idle Timeout: Integer. The unit of this value is seconds. The value is configurable, with range from 0~172800s, step is 60s.

7.13. Error Code

The Error Code is used when the erros happens in HYA network.

The attribute contains the following valueFigure 21.

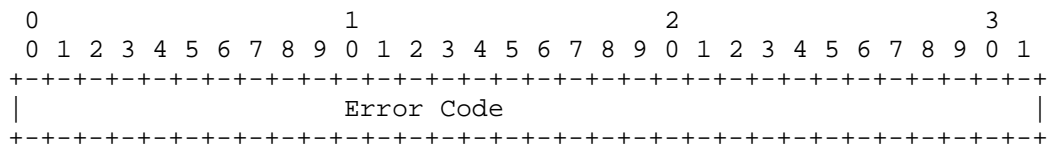


Figure 21: Error Code Attribute

Type: 17 for Error Code Attribute

Length: 4 Bytes

Idle Timeout: Integer. Error cases have to be handled.

7.14. DSL Link Failure

The DSL Link Failure will be used by CPE to inform HAAP of CPE DSL link failure through LTE WAN. Usually, the failure can be detected by HAAP via GRE Hello message. However, it is possible that the local failures happen on CPE. In this case, direct notification to HAAP is a efficiency way than GRE hello mechanism (Failure Detection time is retry times*hello interval)

There is no value in this attribute.

Type: 18 for DSL Link Failure

Length: 0

7.15. LTE Link Failure

The LTE Link Failure will be used by CPE to inform HAAP of CPE LTE link failure through DSL WAN. Usually, the failure can be detected by HAAP via GRE Hello message. However, it is possible that the local failures happen on CPE. In this case, direct notification to

HAAP is a efficiency way than GRE hello mechanism (Failure Detection time is $\text{retry times} \times \text{hello interval}$)

There is no value in this attribute.

Type: 19 for LTE Link Failure

Length: 0

7.16. IPv6 Prefix Assigned to Terminal Host

The IPv6 Prefix assigned to terminal host on CPE will be notified to HAAP. Then HAAP can setup the IPv6 prefix translation mapping between CPE HA IPv6 prefix and the terminal host IPv6 prefix. When the downstream traffic arriving, HAAP can advertise the CPE HA IPv6 prefix for HYA refereed to Section 4.2.

When both DSL link and LTE link are working, CPE will assign BRAS IPv6 prefix to terminal host. When DSL line failure and lead to PPPoE terminated, CPE will assign HAAP IPv6 prefix to terminal host. When DSL line recovers from failure and obtains a new IPv6 prefix from BRAS, CPE will assign BRAS IPv6 prefix to terminal host again. When HG change the IPv6 prefix assigned to terminal host, need to send notify to HAAP.

The attribute contains the following valueFigure 22:

```

      0               1               2               3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
\               IPv6 Prefix Assigned to Terminal Host               \
+-----+-----+-----+-----+-----+-----+-----+-----+
| Network Mask |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Figure 22: IPv6 Prefix Assigned to Terminal Host Attribute

Type: 21 for IPv6 Prefix Assigned to Terminal Host Attribute

Length: 17 Bytes

Value: The first 16 bytes are the IPv6 prefix, the last byte indicates the network mask.

7.17. Subscribed DSL Upstream BW

The Subscribed DSL Upstream BW is used by HAAP to notify CPE the available DSL upstream Bandwidth. The subscribed DSL Upstream BW can be obtained by HAAP during authentication and authorization process. CPE/HAAP will use this value to set the token bucket on DSL for traffic overflow referred to Section 5.4.

The attribute contains the following valueFigure 23

```

      0               1               2               3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|               Subscribed DSL Upstream BW               |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Figure 23: Subscribed DSL Upstream BW

Type: 22 for Subscribed DSL Upstream BW

Length: 4 Bytes

Subscribed DSL Upstream BW: The conventional DSL upstream BW is provided by operator for CPE. The unit of this value is kbps.

7.18. Subscribed DSL Downstream BW

The Subscribed DSL Downstream BW is used by HAAP to notify CPE the available DSL downstream Bandwidth. The subscribed DSL Downstream BW can be obtained by HAAP during authentication and authorization process. CPE/HAAP will use this value to set the token bucket on DSL for traffic overflow referred to Section 5.4.

The attribute contains the following valueFigure 24:

```

      0               1               2               3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|               Subscribed DSL Downstream BW               |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Figure 24: Subscribed DSL Downstream BW

Type: 23 for Subscribed DSL Downstream BW

Length: 4 Bytes

Subscribed DSL Downstream BW: The conventional DSL downstream BW is provided by operator for CPE. The unit of this value is kbps.

7.19. Delay Difference Threshold Violation

The Delay Different Threshold Violation is used to carry the times when the RTT/delay difference exceeds the threshold defined in Figure 16. This times will impact the decision to switch the traffic to DSL GRE tunnel only. When the RTT/delay difference exceeds the threshold above the times defined in this attribute, all the traffic will be switched to DSL tunnel, rather than LTE tunnel. This is the configuration to CPE by HAAP.

The attribute contains the following valueFigure 25.

```

      0               1               2               3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|           Delay Difference Threshold Violation Times           |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Figure 25: Delay Difference Threshold Violation

Type: 24 for Delay Difference Threshold Violation Attribute

Length: 4 Bytes

Delay Difference Threshold Violation Times: The times when the RTT/delay difference exceeds the threshold defined in Figure 16. This value can be configured by operators.

7.20. Delay Difference Threshold Compliance

The Delay Different Threshold Compliance is used to carry the times when the RTT/delay difference stays below the threshold defined in Figure 16. This times will impact the decision to switch the traffic to DSL GRE tunnel only. When the RTT/delay difference stays below the threshold above the times defined in this attribute, all the traffic can be transmitted over HYA, with both LTE and DSL tunnel. This is the configuration to CPE by HAAP.

The attribute contains the following valueFigure 26.

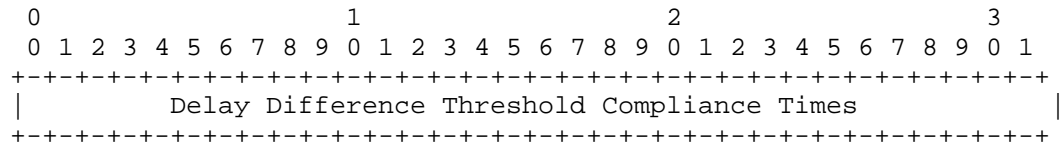


Figure 26: Delay Difference Threshold Compliance

Type: 25 for Delay Difference Threshold Compliance Attribute

Length: 4 Bytes

Delay Difference Threshold Compliance Times: The times when the RTT/delay difference stays below the threshold defined in Figure 16. This value can be configured by operators.

7.21. Filter list ACK

The Filter List ACK attribute is defined for acknowledgement of filter list notify and filter list error notification. This attribute is used as a reply for Figure 14.

The attribute contains the following value Figure 27.

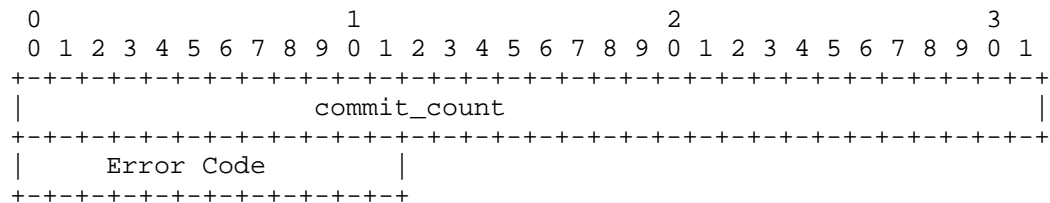


Figure 27: Filter List ACK Attribute

Type: 30 for Filter List ACK Attribute

Length: 5 Bytes

Commit_count: The first 4 bytes is committed count, to differentiate filter list packages in case of change of the filter list package.

Error Code: Code 0 is ACK; code 1 is NACK and indicates this is new dial-in subscriber, which means HAAP should teardown this user to let this user to redial; code 2 is NACK and indicates this is a existing subscriber, HAAP should sent the filter list package to this subscriber again.

7.22. End AVP

The End AVP is used to indicate that this is the last attribute contained in the GRE control messages. There is no value in End AVP.

Type: 255 for End AVP

Length: 0

8. GRE Tunnels State Machine

The following state diagram (Figure 28) represents the life cycle of HYA bonding tunnel.

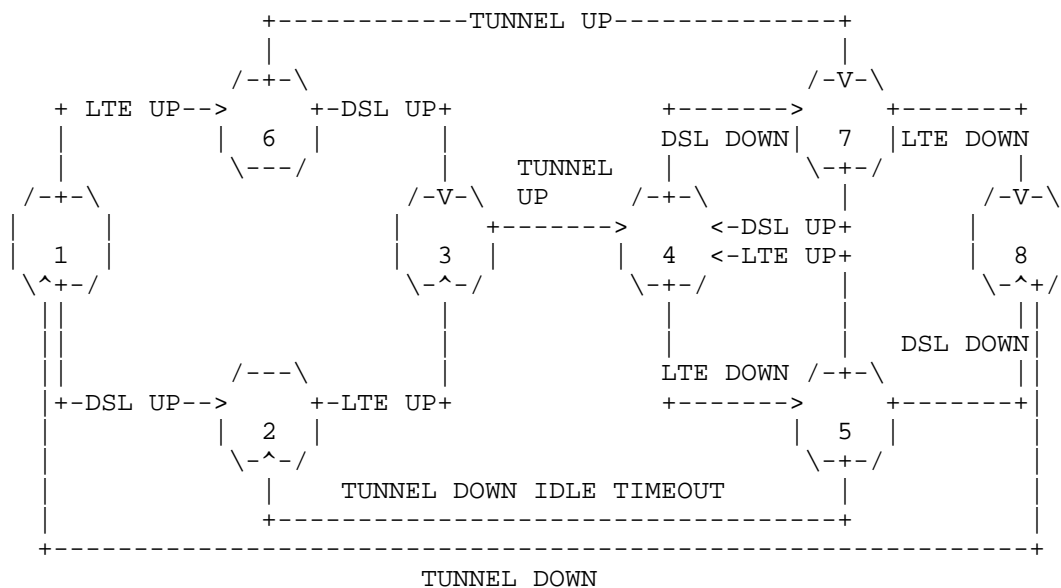


Figure 28: GRE State Machine

The various states are described as below:

State No. =====	DSL Tunnel =====	LTE Tunnel =====	Bonding Tunnel =====
1	Down	Down	Down
2	Up	Down	Down
3	Up	Up	Down
4	Up	Up	Up
5	Up	Down	Up
6	Down	Up	Down
7	Down	Up	Up
8	Down	Down	Up

Tunnel / GRE States

9. IANA Considerations

IANA is requested to allocate one code TBD for the dynamic GRE protocol.

10. Security Considerations

In the whole processing of HA, security of control messages MUST be guaranteed. The CPE discovers the HAAP by resolving the HAAP address over DNS. This protects the CPE against connections to foreign HAAP, if the DNS service and the domain name in the CPE isn't corrupted.

The CPE should be prevented against receiving GRE notifications without a valid session. In the whole processing of end to end HAAP session establishing and GRE notification signaling, the source IP address for session establishment from CPE MUST be strictly verified, including IP address authentication and identification at the HAAP side. Any authentication mechanism with credential or checking the IP address is feasible.

GRE notification key poisoning Every new session at the HAAP generates a magic number, which is encapsulated in the key field of the GRE header and will be carried in the signalling messages and data traffic for verification by comparing the Magic Number in the message and the Magic Number in the local session table. Traffic without a valid Magic Number and outer IP address will be discarded on the HAAP. Magic number is used for both control message and data message security.

For data traffic security, it is also proposed to use IP address validation to protect against IP Spoofing attacks.

11. Acknowledgements

Many thanks to Dennis Kusidlo.

12. Normative References

- [I-D.lhwxz-hybrid-access-network-architecture]
Leymann, N., Heidemann, C., Wasserman, M., and D. Zhang,
"Hybrid Access Network Architecture", draft-lhwxz-hybrid-
access-network-architecture-00 (work in progress), June
2014.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2561] White, K. and R. Moore, "Base Definitions of Managed
Objects for TN3270E Using SMIV2", RFC 2561, April 1999.
- [RFC2697] Heinanen, J. and R. Guerin, "A Single Rate Three Color
Marker", RFC 2697, September 1999.
- [RFC2698] Heinanen, J. and R. Guerin, "A Two Rate Three Color
Marker", RFC 2698, September 1999.
- [RFC2890] Dommety, G., "Key and Sequence Number Extensions to GRE",
RFC 2890, September 2000.
- [RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless
Address Autoconfiguration", RFC 4862, September 2007.
- [TS23.401]
"3GPP TS23.401, General Packet Radio Service (GPRS)
enhancements for Evolved Universal Terrestrial Radio
Access Network (E-UTRAN) access", September 2013.

Authors' Addresses

Nicolai Leymann
Deutsche Telekom AG
Winterfeldtstrasse 21-27
Berlin 10781
Germany

Phone: +49-170-2275345
Email: n.leymann@telekom.de

Cornelius Heidemann
Deutsche Telekom AG
Heinrich-Hertz-Strasse 3-7
Darmstadt 64295
Germany

Phone: +4961515812721
Email: heidemannc@telekom.de

Margaret Wesserman
Painless Security

Email: mrw@painless-security.com

Li Xue
Huawei
NO.156 Beiqing Rd. Z-park, Shi-Chuang-Ke-Ji-Shi-Fan-Yuan
Beijing, HaiDian District 100095
China

Email: xueli@huawei.com

Dacheng Zhang
Huawei
NO.156 Beiqing Rd. Z-park, Shi-Chuang-Ke-Ji-Shi-Fan-Yuan
Beijing, HaiDian District 100095
China

Email: zhangdacheng@huawei.com

Interdomain Routing Working Group
Internet-Draft
Intended status: Informational
Expires: December 27, 2014

N. Leymann
C. Heidemann
Deutsche Telekom AG
M. Wesserman
Painless Security
X. Xue
D. Zhang
Huawei
June 25, 2014

Hybrid Access Network Architecture
draft-lhwz-hybrid-access-network-architecture-00

Abstract

In practice, people have realized that it may be difficult to update or rebuild existing copper networks when they are deployed in certain areas. At the same time, the requirements of customers on bandwidth are continually increased. This document tries to discuss the general network architectures which could be used to address this problem by bundling multiple hybrid access networks together according to the certain management policies.

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]

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 December 27, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	2
2. Terminology	3
3. Motivation Scenario	3
4. Flow-Based Forwarding versus Packet-Based Forwarding	4
5. An Architecture for Packet-Based HYA	6
6. Existing Technologies and Gap Analysis	8
7. Security Considerations	9
8. Acknowledgements	9
9. Normative References	9
Authors' Addresses	9

1. Introduction

It could be difficult for operators to upgrade or rebuild their copper access networks deployed in certain places (e.g., the old downtown areas). However, at the same time, the requirements of customers on broader bandwidth become stronger. To address this problem, the possibility of combining different or hybrid access networks (e.g., LTE and DSL) for a higher bandwidth is being discussed.

To achieve this functionality, the mechanism for binding multiple hybrid access networks need to be designed, which is called as HYbrid access (HYA) mechanism in this document. A HYA mechanism may need to have the capability in flexibly deciding the paths to forward data traffics. This document attempts identify the potential issues and requirements related with the HYA mechanisms and proposes general architectural design suggestions.

The remainder of this document is organized as follows. Section 2 lists the key terms used in this document. Section 3 introduces a

motivation scenario and requirements in combining hybrid access networks. Section 4 discusses the criteria of identifying the packet forwarding paths between the combined hybrid access networks. In section 5, a general HYA architecture is proposed for constructing the packet-based forwarding solutions. Section 6 discusses the possibility of using existing multi-path technologies in addressing the HYA issues and tries to identify the gaps.

2. Terminology

Customer Premise Equipment (CPE): A device that connects multiple hosts to provide connectivity to the service providers network.

HYbrid Access (HYA): HYbrid Access (HYA) is the bundling of two or more access lines over different technologies (e.g. DSL and LTE) to one Internet connection for end customers.

Hybrid Access Aggregation Point (HAAP): The HAAP which acts as a service termination and a service creation implements bonding mechanism and sets up a high speed Internet dual stack IP connection with CPE on top of two or more hybrid access technologies. The packet reorder, reassemble functions in packet-based solutions should be supported on HAAP.

Path: A sequence of links between the CPE and HAAP, typically DSL path and LTE path are defined in this document.

3. Motivation Scenario

The figureFigure 1 illustrates a motivation scenario, in which a customer accesses the Internet through a DSL access Network. The requirements of the customer on broader bandwidth for better service experience become stronger. However, the bandwidth of the DSL access network has been fully occupied (i.e., the traffics on the copper line has reached to a pre-specified threshold) and cannot satisfy any further bandwidth requirements from the customer. In addition, because the customer is located in an old downtown street, it may take a long time or be extremely difficult for the operator to get the official construction permit to update the DSL access network or deploy a new one in that area. Whereas, at the same time, the operator has already deployed a LTE backhaul network in the downtown area which is still not used to its fullest. If the operator is able to take advantage of the bandwidth resources of the LTE and DSL network to transfer the traffics of the customer concurrently, it is possible to provide a higher bandwidth to the customer and guarantee good customer experiences.

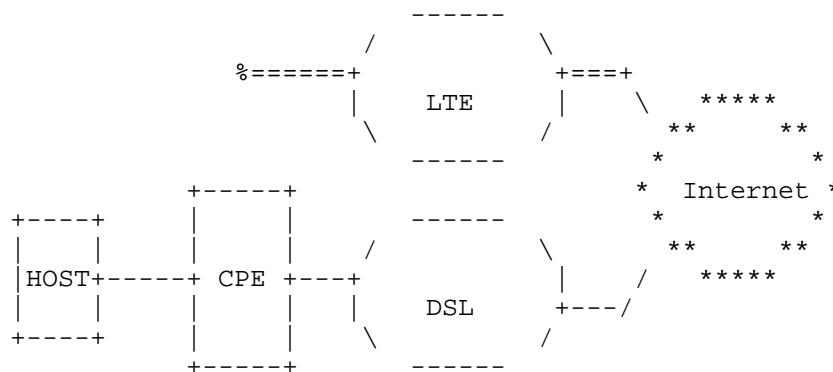


Figure 1: Existing Home Network Scenario

As illustrated in Figure 2, in order to bind the DSL and LTE access networks, the Customer Premise Equipment (CPE) of the customer's home network should have at least two Wide Area Network (WAN) interfaces (noted as E and D in Figure 2) for connecting the LTE and DSL access networks respectively. The network architecture proposed in Figure 2 could be extended if there are other access networks available for the combination.

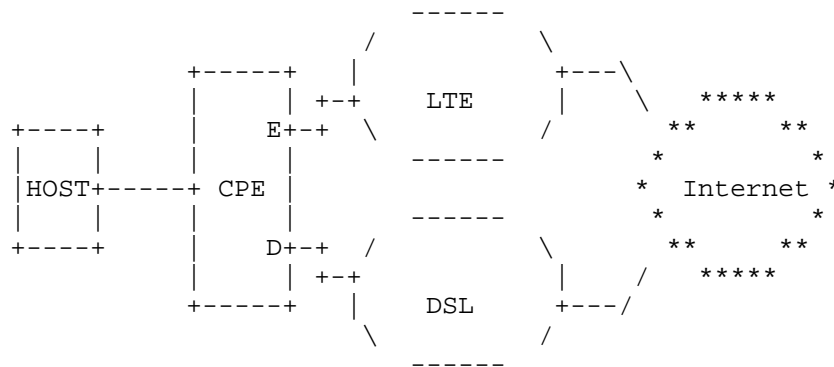


Figure 2: Hybrid Access Scenario

4. Flow-Based Forwarding versus Packet-Based Forwarding

According to the criteria of identifying the packet forwarding paths, HYA mechanisms can be classified into flow-based HYA mechanisms and packet-based HYA mechanisms.

In a flow-based mechanism, customer traffics are broken into data flows, each of which is associated to a single forwarding path

Figure 3. The packets of a certain flow can be identified by, for instance, its destination address, source address, or 5-tuple IP parameters, etc. Upon on receiving a packet from the hosts, the CPE device will identify the flows that the packet belongs to and forward the packet according to the pre-specified policies, such as flow A is distributed into LTE path and flow B is distributed into DSL path.

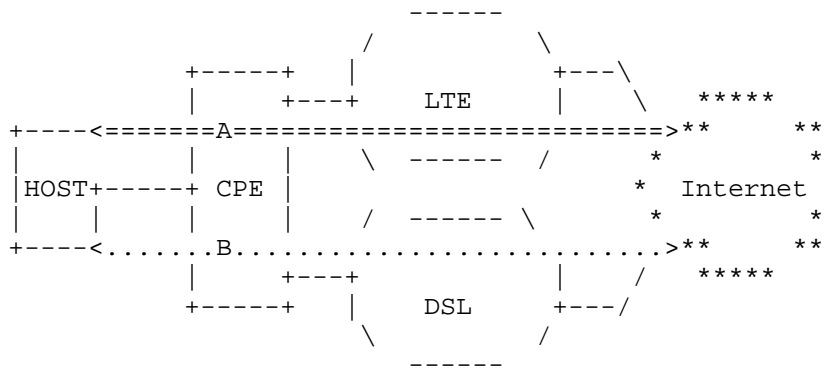


Figure 3: Flow-Based Forwarding

Flow-based distribution is very similar to load balance technologies and easy for operator to deploy. On the other side, the disadvantages of flow-based solutions are obvious. The bandwidth consumption of each flow could change over time and it could be difficult to predict. Thus, the traffic balance between the different paths is difficult to guarantee. In addition, in certain scenarios, it may be difficult to guarantee the upstream and downstream packets within the same flow are transferred in the same data path.

For instance, according to pre-specified policies, a CPE needs to select a flow and forward the packets within the flow through the LTE network when the overload of the DSL network reaches a per-specified threshold. However, the bandwidth consumption of the flow associated with the LTE network becomes big later and causes the congestion of LTE work. A more detailed gap analysis for flow-based solutions will be provided in the next version of this document.

In a packet-based solution, instead, the forwarding policies are specified at the packet level. A CPE can flexibly decide which packets should be forwarded through the LTE access network when the DSL network is heavily loaded. Each packet is associated to a single forwarding path while different packets belonging to the same flow could be transferred by different pathsFigure 4. Therefore, compared to flow-based solutions, the CPE in a packet-based solution can tune

the bandwidth consumption on different paths in a flexible and fine-grained way.

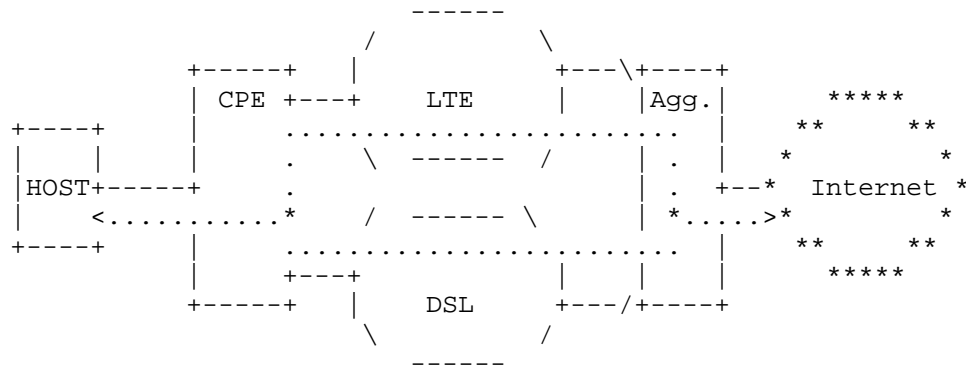
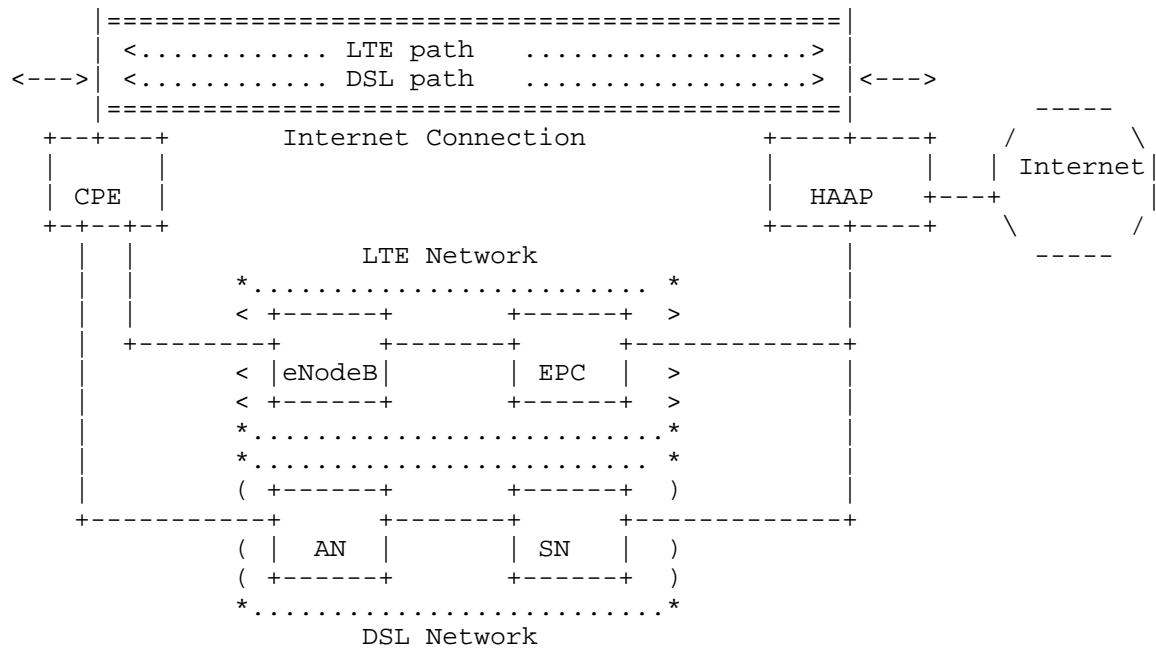


Figure 4: Packet-Based Forwarding

In packet-based solutions, due to different transporting delivery caused by LTE and DSL paths, the packets in the same flow may reach their destination in different orders. It could be desired to provide a device (see the Agg in Figure 4) to perform traffic reordering and reassembling at the remote side. In a flow-based solution, the out-of-order packet issues will not occur in the upstream traffics, while it may occur in the downstream packets.

5. An Architecture for Packet-Based HYA

An architecture for packet-based HYA mechanisms with packet-based distribution is illustrated in Figure 5. In the diagram, an endpoint (Hybrid Access Aggregation Point (HAAP)) is deployed at the remote side of the CPE and carries out the packet reordering and reassembling functions. Only if the utilization of DSL bandwidth has reached to a pre-specified threshold, CPE and HAAP would distribute customer traffic on packet-based between DSL and LTE path.



Legend:

AN Access Node
 SN Service Node
 EPC Evolved Packet Core

Figure 5: Hybrid Access Network Architecture

A full-fledged packet-based HYA mechanism using this architecture should meet following several requirements:

1. **Network Agnostic:** On the client side, the CPE must implement the bond mechanism and distribute the customer traffic between these two interfaces based on per-packet. On the network side, an endpoint HAAP cooperates with the CPE to achieve packet reorder, packet reassemble functions etc. The HYA connection is only terminated and managed at the CPE and the HAAP. Therefore either the DSL and LTE network infrastructure are not changed and impacted.
2. **Path Management:** As a result of successful authentication, the CPE needs to negotiate with HAAP so as to setup and manage the HYA connection dynamically through the DSL and LTE physical paths. Additionally, the bundle two paths may have different characteristics such as rate, delay or MTU etc. A mechanism of path management should also fix this gap.

3. Traffic Overflow Function: In order to guarantee the cheapest path used first, the CPE need to get the downstream and upstream DSL bandwidth from the network, and periodically check the bypass bandwidth and notify the result to the HAAP. Based on the negotiation, HAAP can adjust the threshold of the DSL path and adapt the packet-based routing decision dynamically.
 4. Backward Compatibility: In order to ensure that existing services are not influenced by HYA architecture, certain traffic must not be routed through HYA connection, but directly over the specific interface. The negotiation between HG and HAAP for this policy routing must be defined.
6. Existing Technologies and Gap Analysis

There are various technologies (e.g., MPTCP[RFC6182] , MLPPP[RFC1990]) which enable to similar requirements to support the simultaneous use of multiple data paths.

In MPTCP, the primary use case is to support application layer for the simultaneous use of multiple path between the multihomed hosts. It needs to analysis and consider the issues with various middleboxes impaction. For example, MPTCP falls back to ordinary TCP if a middlebox alters the payload. For HYA architecture in network layer, these mechanisms are overload. By far, the MPTCP does not support packet-based distribution requirement between the multiple path specified in Section 5. Therefore, only fair-share is supported by MPTCP, MPTCP does not meet the traffic overflow function specified in Section 5. For backward compatibility, MPTCP can not recognize the IP layer information and consequently have issues to support existing traffic unimpaired requirement.

In MLPPP, the link-layer protocol (PPP[RFC1990]) is extended to combine multiple PPP link. The primary scenario is for fragmented protocol data units (PDU) on link layer to be transferred on multiple link and be reassembled back into the original PDU. By far, the MLPPP does not apply to the HYA deployment scenario, which is IP network between CPE and HAAP. Moreover, MLPPP does not meet the requirements as packet-based distribution between the multiple path and traffic overflow function specified in Section 5. For backward compatibility, MLPPP can not recognize the IP layer information and consequently have issues to support existing traffic unimpaired requirement as MPTCP.

7. Security Considerations

tbd

8. Acknowledgements

Many thanks to Dennis Kusidlo.

9. Normative References

- [RFC1990] Sklower, K., Lloyd, B., McGregor, G., Carr, D., and T. Coradetti, "The PPP Multilink Protocol (MP)", RFC 1990, August 1996.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC6182] Ford, A., Raiciu, C., Handley, M., Barre, S., and J. Iyengar, "Architectural Guidelines for Multipath TCP Development", RFC 6182, March 2011.

Authors' Addresses

Nicolai Leymann
Deutsche Telekom AG
Winterfeldtstrasse 21-27
Berlin 10781
Germany

Phone: +49-170-2275345
Email: n.leymann@telekom.de

Cornelius Heidemann
Deutsche Telekom AG
Heinrich-Hertz-Strasse 3-7
Darmstadt 64295
Germany

Phone: +4961515812721
Email: heidemannc@telekom.de

Margaret Wesserman
Painless Security

Email: mrw@painless-security.com

Li Xue
Huawei
NO.156 Beiqing Rd. Z-park, Shi-Chuang-Ke-Ji-Shi-Fan-Yuan
Beijing, HaiDian District 100095
China

Email: xueli@huawei.com

Dacheng Zhang
Huawei
NO.156 Beiqing Rd. Z-park, Shi-Chuang-Ke-Ji-Shi-Fan-Yuan
Beijing.Haidian District 100095
China

Email: zhangdacheng@huawei.com

HOMENET
Internet-Draft
Intended status: Standards Track
Expires: April 24, 2014

D. Migault (Ed)
Orange
October 21, 2013

DNSSEC Validators DHCP Options
draft-mglt-homenet-dnssec-validator-dhc-options-02.txt

Abstract

DNSSEC provides data integrity and authentication for DNSSEC validators. However, without valid trust anchor(s) and an acceptable value for the current time, DNSSEC validation cannot be performed. As a result, there are multiple cases where DNSSEC validation MUST NOT be performed. In addition, this list of exceptions is expected to become larger over time.

Considering an increasing number of cases where DNSSEC is disabled adds complexity to the DNSSEC validator implementations and increases the vectors that disable security.

This document assumes that DNSSEC adoption by end devices requires that end devices MUST be able to support a DNSSEC validation always set. This MUST be valid today as well as in the future.

This document describes DHCP Options to provision the DHCP Client with valid trust anchors and time so DNSSEC validation can be performed.

Status of This Memo

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

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

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

This Internet-Draft will expire on April 24, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Requirements notation	2
2. Introduction	2
3. Threat Model	3
3.1. Motivations for providing DNSSEC Trust Anchor	3
3.2. Motivations for providing Time	5
4. Terminology	5
5. DHCP DNSSEC Trust Anchor Options	6
5.1. DHCP DNSSEC KSK RR Trust Anchor Options	6
5.2. DHCP DNSSEC KSK CERT Trust Anchor Options	6
6. DHCP Time Option	7
7. DHCP Client Behavior	8
8. DHCP Server Behavior	9
9. DHCP Relay Agent Behavior	10
10. IANA Considerations	10
11. Security Considerations	10
12. Acknowledgment	10
13. References	10
13.1. Normative References	10
13.2. Informational References	11
Appendix A. Document Change Log	12
Author's Address	12

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

DNSSEC [RFC4033], [RFC4034], [RFC4035] adds data authentication and integrity checks to DNS [RFC1034], [RFC1035]. For signature validation, DNSSEC requires a trust anchor such as the Key Signing

Key (KSK) of the Root Zone or any other zone. Without a trust anchor, DNSSEC validation cannot be performed. In addition KSKs and signatures are valid for a given period of time. As a result, DNSSEC validation cannot be performed if time shifting is too large.

This document considers DHCP DNSSEC Trust Anchor Option and DHCP Time Option to provision a device with trusted KSKs and current time. Although our priority is to provide the Root Zone KSK, we also consider the case other trusted KSK MAY be provided, for example, if a Zone does not provide secure delegation, or to mitigate badly configured DNSSEC zones (like TLDs zones).

The main motivation for these DHCP Options is that DHCP enabled devices have DNSSEC validation always set and do not need to perform DNS resolution without DNSSEC validation. In fact, enabling DNS with no validation represents a potential way to remove security and MAY be used by attackers. Similarly, DNSSEC configuration implemented in the end users device, MAY not consider future cases and MAY introduce vulnerabilities. DHCP Options prevent this as long as the relationship between DHCP Client and DHCP Server is trusted.

This document assumes that the channel between the DHCP Client and the DHCP Server is trusted and secured with DHCP mechanisms described in [RFC3315], or IPsec [RFC4301].

3. Threat Model

This document addresses the case of a device configured with DNSSEC validation set that is plugged in, gets connectivity (using DHCP for example), but fails DNSSEC resolutions because its trust anchor KSK is not valid anymore or its local time is not valid.

This threat mainly addresses devices that can be switched off for a long period of time or devices that MAY be off-shelves for a long time before being plugged in. CPEs as well as any homenet devices are concerned by this use case.

This threat also addresses DNSSEC emergency key roll over operations. Devices that have cached the out-of-date KSK will not be able to check the signatures until the TTL has expired on all caches.

This document proposes DHCP Options that provide the necessary parameters to perform DNSSEC validation. These Options MUST be used on a trusted network over a trusted channel between the DHCP Client and the DHCP Server. These options MAY be used in conjunction of additional mechanisms.

3.1. Motivations for providing DNSSEC Trust Anchor

The first motivation for providing trusted KSKs is to provide automatic configuration of devices to enable DNSSEC validation. This avoids validator initial KSK provisioning issue as well as KSK roll over issues.

A validator MAY not be able to perform signature check with an authenticated KSK because:

- 1) It does not have a trust anchor (like the Root Zone KSK)
- 2) The KSK MAY have been authenticated, stored or cached with an expiration date valid but is not valid anymore. This MAY happen in the case of an emergency key roll over, if the device has been offline during the key roll over, or if the key roll over is not performed as described in [DPS-KSK], [RFC5011].
- 3) The chain of trust MAY have been broken. This can happen to non Root Zone KSK only and MAY not involve the responsibility of the owner of the zone. The deeper the Zone is in the hierarchy, the more likely this happens.
- 4) A DNSSEC zone MAY have been badly signed or a KSK MAY have been badly generated. The DNSSEC MAY be correct, but DNSSEC validator MAY keep for a long time the badly generated KSK, ZSK...

The goal of the DHCP DNSSEC Trust Anchor Option is to provide these validators trusted anchors like the Root Zone KSK, as well as other KSKs (TLDs...) so the validator has the proper KSKs to perform DNSSEC validation.

Most documents are currently focused on the Root Zone KSK for which recommendations and alternative mechanisms have been described. [I-D.jabley-dnsop-validator-bootstrap] provides guide lines on how to retrieve and select DNSSEC Trust Anchors. Section 5.3 and [I-D.jabley-dnssec-trust-anchor] describes mechanisms to retrieve securely the Root Zone KSK relying on TLS security. It suggests to use insecure DNS resolution to set HTTPS connections. Using HTTPS requires downloading the keyDigest id (key-label) from <https://data.iana.org/root-anchors/root-anchors.xml>, followed by an HTTPS request at <https://data.iana.org/root-anchors/key-label.crt> to get the whole certificate.

The key advantages of the DHCP DNSSEC Trust Anchor Option described in this document are that we extend the mechanism to any KSK, and validators can set DNSSEC validation for all DNS queries. However, we do not see any contradiction between recommendations provided by [I-D.jabley-dnsop-validator-bootstrap] and [I-D.jabley-dnssec-trust-

anchor] and believe the principle described in these documents SHOULD be applied by the validators. Note also that DHCP DNSSEC Trust Anchor Option only benefits to validators that are configured via DHCP.

To recover from a DNSSEC failure and remove a particular data from cache, [I-D.jabley-dnsop-dns-flush] suggests to use a NOTIFY message between Authoritative Servers and Resolvers. This mechanism is set between Recursive Server and Authoritative Servers with a specific trusted relationship. This is probably a selection of TLDs. This document, does not address the DNSSEC failure over Recursive Servers, but addresses more specifically DHCP configured devices. These are typically CPEs or End Users. We believe that configuring and restarting DNSSEC validators with DHCP Option, is an easier way to cope with this issue. First the trust relation between DHCP Server already exists, we do not need additional trusted channel between Authoritative Servers or eventually the Recursive Servers. Then basic implementations of stub resolvers, in CPE or desktops may not address NOTIFY message.

3.2. Motivations for providing Time

KSKs and signatures are always associated to an expiration time. As a result, DNSSEC validation requires that the validator knows the current time.

A number of mechanisms exists like [TSLDATE] or [RFC5905] for setting the time of the device. In addition, [RFC5908] provides a Network Time Protocol (NTP) Server Option for DHCP. The DHCP Time Option described in this document differs from [RFC5908] as it provides an estimation of the current time, instead of providing the NTP servers location information. The time value provided by the DHCP Time Option should be used only if previously mentioned mechanisms are either not implemented on the device or are unavailable. One of the reason MAY be that you MAY need valid DNS(SEC) resolution to use these protocols. The time provided by the DHCP Time Option does not have the accuracy of NTP and SHOULD be considered as a best effort value. [I-D.jabley-dnsop-validator-bootstrap] also recommends that when time has not been verified by the validator, the signature validation SHOULD be done with time off.

The key advantage of the DHCP Time Option is that it makes possible to have DNSSEC validation always set. It limits the possible DNSSEC validation variants which potentially expose the device to disable DNSSEC validations. Note also that DHCP Time Option only benefits to validators that are configured via DHCP.

4. Terminology

5. DHCP DNSSEC Trust Anchor Options

This section describes two options:

- DHCP DNSSEC KSK Trust Anchor Options: carries the KSK RRset as described in [RFC1035] with a DNSKEY RDATA as described in [RFC4033]. This data is not integrity protected, nor it can be authenticated. Such data SHOULD be trusted over a trusted DHCP channel.
- DHCP DNSSEC CERT Trust Anchor Options: Carries a certificate encoded as described in [RFC4398]. The advantage of the Certificate is that it enables authentication of the received information by a trusted party. For example, CPE providers MAY provide a trusted certification authority. Unlike DNSSEC key roll over, the CPE provider controls the key roll over of the certification authority it provides.

5.1. DHCP DNSSEC KSK RR Trust Anchor Options

The DHCP DNSSEC KSK Trust Anchor Option provides the RRset as mentioned in the DNS(SEC) Zone. In other words, it carries the RR as defined in Section 3.2. of [RFC1035] and a RDATA DNSKEY as defined in Section 2.1 of [RFC4033]. As the RR has a variable length, the DHCP DNSSEC KSK Trust Anchor Options follows the recommendation format of Section 5.9 of [I-D.ietf-dhc-option-guidelines].

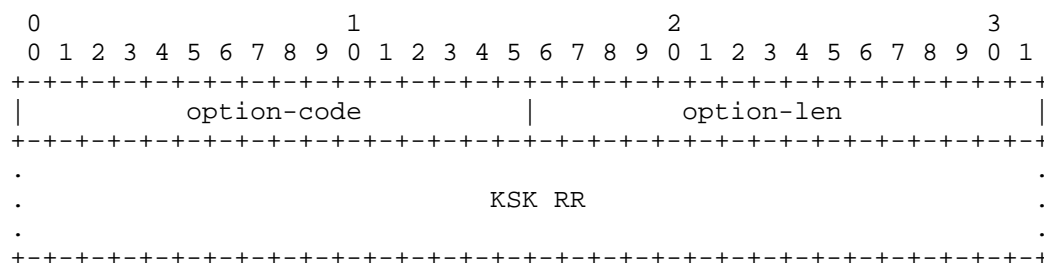


Figure 1: DHCP DNSSEC KSK Trust Anchor Options
Payload Description

- option-code: OPTION_DNSSEC_KSK_RR_TRUST_ANCHOR
- option-len: An unsigned integer giving the length of the KSK RR field in this option in octets

5.2. DHCP DNSSEC KSK CERT Trust Anchor Options

The DHCP DNSSEC CERT Trust Anchor Option provides a certificate. The CERT RR is described in [RFC4398]. Note that only the RDATA associated to the CERT is present in the DHCP Option. As the RR has a variable length, the DHCP DNSSEC KSK CERT Trust Anchor Options follows the recommendation format of Section 5.9 of [I-D.ietf-dhc-option-guidelines].

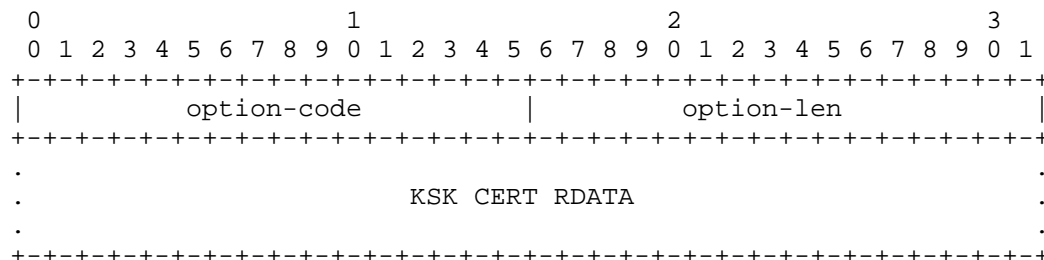


Figure 2: DHCP DNSSEC CERT Trust Anchor Options
Payload Description

- option-code: OPTION_DNSSEC_CERT_TRUST_ANCHOR
- option-len: An unsigned integer giving the length of the KSK RR field in this option in octets

The X.509 [RFC5280] certificate MUST have a keyUsage set to digitalSignature (0) and nonRepudiation (1). Subject Alternative Name DNS name indicates the name of the zone.

In order to be compliant with the certificate of the Root Zone described [I-D.jabley-dnssec-trust-anchor]. The CERT for a KSK SHOULD have a Common Name (CN) with the string "'Zone-FQDN' Zone KSK" followed by the time and date of key generation in the format specified in [RFC3339]. 'Zone-FQDN' is the name of the zone and SHOULD be the same as the one mentioned in Subject Alternative Name. The resourceRecord Attribute SHOULD be set with the DS RRset.

6. DHCP Time Option

The DHCP DNSSEC Time Option is used by the DHCP Server to indicate the Time to the DHCP Client. The Time is provided in a string format as specified in [RFC3339] and in [I-D.ietf-dhc-option-guidelines] Section 5.8.

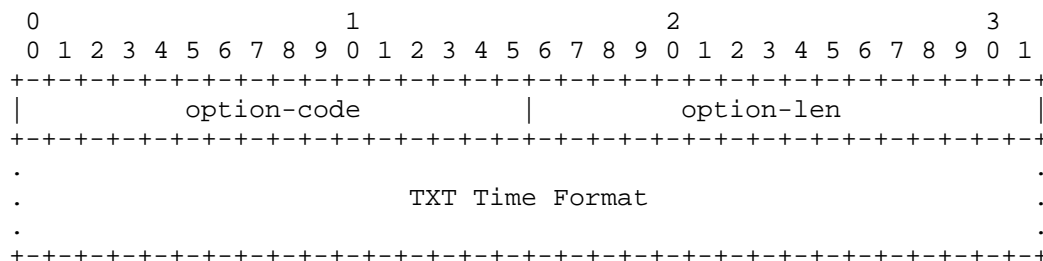


Figure 2: DHCP Time Options
Payload Description

- option-code: OPTION_TIME
- option-len: A string representing the Time

7. DHCP Client Behavior

DHCP DNSSEC KSK Trust Anchor Option, DHCP DNSSEC CERT Trust Anchor Option or DHCP Time Option described in this document are intended for DNSSEC validation. If a connected device is not performing DNSSEC validation, it MUST NOT send a DHCP an Option Request DHCP Option (ORO) [RFC3315] for any of these options, and MUST ignore all these options if provided by the DHCP Server.

The DHCP sends a DHCP ORO for one or multiple options described in the document. Motivations for sending this Option Request DHCP Option is out of scope of the document. It could be a device switched off for a long time, a device that cannot validate the DNSSEC responses.

A channel is considered trusted if 1) the DHCP Server is trusted and authenticated and 2) exchanged data between the DHCP Client and the DHCP Server is integrity protected. IPsec [RFC4301], for example, MAY be used to establish a secure channel.

Over a trusted channel, the DHCP Client that performs DNSSEC validation MAY send an ORO for any of the DHCP DNSSEC KSK Trust Anchor Option, the DHCP DNSSEC CERT Trust Anchor Option or the DHCP Time Option to a DHCP Server.

Over a trusted channel, the DHCP Client that performs DNSSEC validation SHOULD consider the DHCP DNSSEC KSK Trust Anchor Option, the DHCP DNSSEC CERT Trust Anchor Option or the DHCP Time Option sent by the DHCP Server.

Over a non trusted channel, the DHCP Client MAY only send ORO for a DHCP DNSSEC CERT Trust Anchor Option. This option is the only one that MAY be considered by the DHCP Client if sent by the DHCP Server. If the DHCP Client does not trust the signer of the certificate, the option MUST be ignored.

When a DHCP DNSSEC KSK Trust Anchor Option or a DHCP DNSSEC CERT Trust Anchor Option is accepted by the DHCP Client, it MUST remove overwrite old values for the KSK with the new one.

When a DHCP Time Option is accepted by the DHCP Client, it MUST check the difference between its clock and the time provided by the Option. It SHOULD overwrite its clock value only if the difference is too large.

In any other case, ORO requests MUST NOT be sent by the DHCP Client, and options received by the DHCP Server MUST NOT be considered by the DHCP Client. The remaining of the section details when the options MUST NOT be requested by the DHCP Client and MUST be ignored by the DHCP Client when received by the DHCP Server.

The DHCP Client MUST NOT send an ORO for a DHCP DNSSEC KSK Trust Anchor Option, a DHCP DNSSEC CERT Trust Anchor Option or a DHCP Time Option to a DHCP Server that is either not trusted or not authenticated.

All DHCP DNSSEC KSK Trust Anchor Option, a DHCP DNSSEC CERT Trust Anchor Option or a DHCP Time Option received from DHCP Server that is not authenticated or that is not trusted MUST be ignored by the DHCP Client.

The DHCP Client MUST NOT send an ORO for a DHCP DNSSEC KSK Trust Anchor Option or a DHCP Time Option to a trusted DHCP Server over an untrusted channel. A DHCP DNSSEC CERT Trust Anchor Option MAY be requested over an untrusted channel since the certificate is signed and thus can be authenticated. A DHCP DNSSEC CERT Trust Anchor Option signed by an untrusted authority MUST be ignored by the DHCP Client.

All DHCP DNSSEC KSK Trust Anchor Option or a DHCP Time Option received from DHCP Server over a channel that is not trusted MUST be ignored by the DHCP Client.

8. DHCP Server Behavior

The DHCP Server SHOULD properly answer with the requested options in the ORO, even if the DHCP Server does not consider the channel with DHCP Client as trusted.

The DHCP Server MAY also provide DHCP DNSSEC KSK Trust Anchor Option, DHCP DNSSEC CERT Trust Anchor Option or DHCP Time Option without being requested by the DHCP Client. This could for example prevent failures not detected by the DHCP Client.

9. DHCP Relay Agent Behavior

The DHCP Options described in the document do not impact the Relay Agent.

10. IANA Considerations

The DHCP options detailed in this document is:

- OPTION_DNSSEC_KSK_RR_TRUST_ANCHOR: TBD
- OPTION_DNSSEC_KSK_CERT_TRUST_ANCHOR: TBD
- OPTION_TIME: TBD

11. Security Considerations

Security has been discussed in the "DHCP Client Behavior Section". As information contained in the payloads are use to enable signature validation, these pieces of information MUST be considered only when issued by a trusted party, and when integrity protection is provided.

12. Acknowledgment

Bringing DNSSEC in Home Networks discussion has started during the IETF87 in Berlin with Ted Lemon, Ralph Weber, Normen Kowalewski, and Mikael Abrahamsson. An email discussion has also been initiated by Jim Gettys with among others, helpful remarks from Paul Wouters, Joe Abley, Michael Ridchardson.

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.
- [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.
- [RFC3339] Klyne, G., Ed. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, July 2002.
- [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.
- [RFC4398] Josefsson, S., "Storing Certificates in the Domain Name System (DNS)", RFC 4398, March 2006.
- [RFC5011] StJohns, M., "Automated Updates of DNS Security (DNSSEC) Trust Anchors", STD 74, RFC 5011, September 2007.
- [RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.
- [RFC5905] Mills, D., Martin, J., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, June 2010.
- [RFC5908] Gayraud, R. and B. Lourdelet, "Network Time Protocol (NTP) Server Option for DHCPv6", RFC 5908, June 2010.

13.2. Informational References

- [DPS-KSK] Ljunggren, F., Okubo, T., Lamb, R., and J. Schlyter, "DNSSEC Practice Statement for the Root Zone KSK Operation", Root DNSSEC Design Team, URL: <http://www.root-dnssec.org/wp-content/uploads/2010/06/icann-dps-00.txt>, 2010.

[I-D.ietf-dhc-option-guidelines]

Hankins, D., Mrugalski, T., Siodelski, M., Jiang, S., and S. Krishnan, "Guidelines for Creating New DHCPv6 Options", draft-ietf-dhc-option-guidelines-14 (work in progress), September 2013.

[I-D.jabley-dnsop-dns-flush]

Abley, J., "A Mechanism for Remote-Triggered DNS Cache Flushes (DNS FLUSH)", draft-jabley-dnsop-dns-flush-00 (work in progress), June 2013.

[I-D.jabley-dnsop-validator-bootstrap]

Abley, J. and D. Knight, "Establishing an Appropriate Root Zone DNSSEC Trust Anchor at Startup", draft-jabley-dnsop-validator-bootstrap-00 (work in progress), January 2011.

[I-D.jabley-dnssec-trust-anchor]

Abley, J., Schlyter, J., and G. Bailey, "DNSSEC Trust Anchor Publication for the Root Zone", draft-jabley-dnssec-trust-anchor-07 (work in progress), June 2013.

[TSLDATE] error, IO., "tlsdate: secure parasitic rdate replacement", URL: <https://github.com/ioerror/tlsdate>, 2013.

Appendix A. Document Change Log

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

-00: First version published.

Author's Address

Daniel Migault
Orange
38 rue du General Leclerc
92794 Issy-les-Moulineaux Cedex 9
France

Phone: +33 1 45 29 60 52
Email: mglt.ietf@gmail.com

HOMENET
Internet-Draft
Intended status: Standards Track
Expires: January 5, 2015

D. Migault (Ed)
Orange
W. Cloetens
SoftAtHome
C. Griffiths
Dyn
R. Weber
Nominum
July 4, 2014

Outsourcing Home Network Authoritative Naming Service
draft-mglt-homenet-front-end-naming-delegation-04.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/>.

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

This Internet-Draft will expire on January 5, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Requirements notation	3
2. Introduction	3
3. Terminology	4
4. Architecture Description	5
4.1. Architecture Overview	5
4.2. Example: DNS(SEC) Homenet Zone	7
4.3. Example: CPE necessary parameters for outsourcing	9
5. Synchronization between CPE and Public Authoritative Servers	10
5.1. Synchronization with a Hidden Master	10
5.2. Securing Synchronization	11
5.3. CPE Security Policies	12
6. DNSSEC compliant Homenet Architecture	13
6.1. Zone Signing	13
6.2. Secure Delegation	15
7. Handling Different Views	15
8. Reverse Zone	15
9. Security Considerations	16
9.1. Names are less secure than IP addresses	16
9.2. Names are less volatile than IP addresses	16
10. IANA Considerations	16
11. Acknowledgment	16
12. References	17
12.1. Normative References	17
12.2. Informational References	18

Appendix A. Document Change Log	19
Authors' Addresses	20

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, [I-D.ietf-homenet-arch] recommends that homenet networks be resilient to connectivity disruption from the ISP. This requires that a dedicated 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 the impact of multiple views and Section 8 discusses the case of the reverse 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. The content of the DNS(SEC) Homenet Zone 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`.

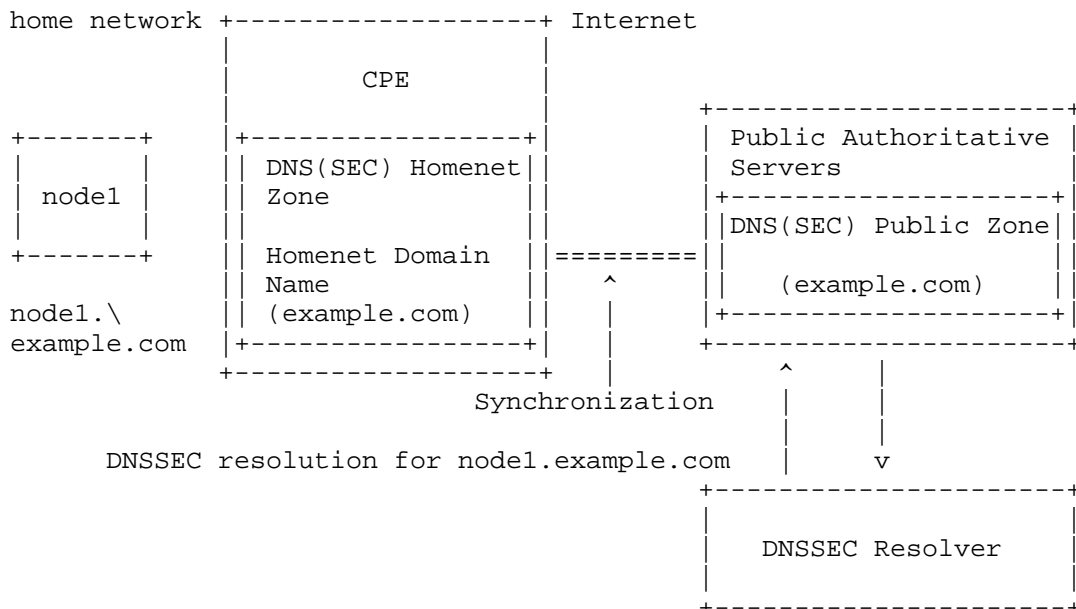


Figure 1: Homenet Naming Architecture Description

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 the one this document considers here is zone signing when the

CPE outsources this operation. Implications of such policy are detailed in Section 6 and Section 7.

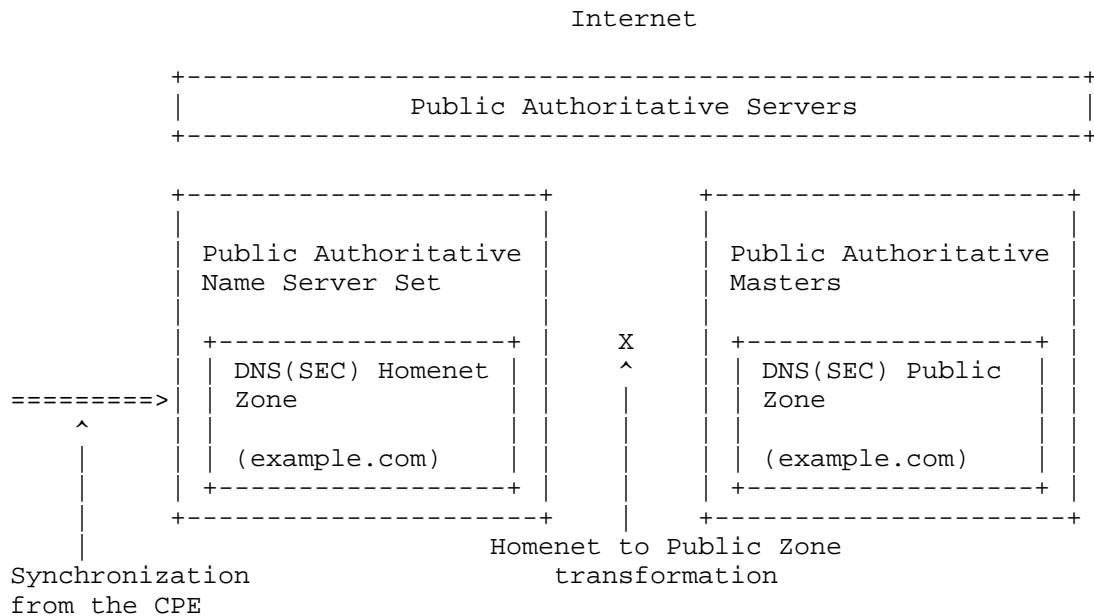


Figure 2: Public Authoritative Servers Description

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]. 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 than a NOTIFY, and they do not provide the server the opportunity to postpone 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 synchronize 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 [RFC5996] 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

[I-D.ietf-homenet-arch] 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. This aspect is discussed in detail in Section 7. More specifically, if the CPE signs the DNS Homenet Zone, then, the CPE has the ability to host the authoritative naming service of the homenet for DNSSEC queries coming from within the network. As a result, a query will be resolved the same way whether it is sent from the home network or from the Internet. On the other hand, if the CPE does not sign the DNS Homenet Zone, either it acts as an authoritative server for the home network or not. If the CPE is an authoritative server for queries initiated

from within the home network, then nodes connected to both networks-- the home network and the Internet -- do not have a unique resolution. Devices that may be impacted are mobile phones with Radio Access Network interfaces and WLAN interfaces. Alternatively if the CPE does not act as an authoritative server, it goes against the principles connectivity disruption independence exposed in [I-D.ietf-homenet-arch] section 4.4.1 and 3.7.5. In case of connectivity disruption, naming resolution for nodes inside the home network for nodes in the home network are not possible.

- 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.mglt-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 but 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, [I-D.ietf-dnsop-delegation-trust-maintainance] may mitigate such issues.

7. Handling Different Views

The issue raised by handling different views of the DNS Homenet Zone or a DNS Homenet Zone that differs from the Public Zone is that a given DNS query may lead to different responses. The responses may be different values for the queried RRsets or different RCODE or different RRsets types in the responses for DNS/DNSSEC responses.

The document does not recommend the CPE manages different views, since devices may be connected to different networks at the same time or may flip / flop from one network to the other.

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.mglt-homenet-naming-architecture-dhc-options] describes how automatic configuration may be performed.

9. Security Considerations

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

9.1. 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.

9.2. 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 than 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.

10. IANA Considerations

This document has no actions for IANA.

11. 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.

12. References

12.1. 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.
- [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.

- [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.
- [RFC5996] Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5996, September 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.

12.2. Informational References

- [I-D.ietf-dnsop-delegation-trust-maintainance]
Kumari, W., Gudmundsson, O., and G. Barwood, "Automating DNSSEC Delegation Trust Maintenance", draft-ietf-dnsop-delegation-trust-maintainance-14 (work in progress), June 2014.
- [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.mglt-homenet-naming-architecture-dhc-options]
Migault, D., Cloetens, W., Griffiths, C., and R. Weber, "DHCP Options for Homenet Naming Architecture", draft-mglt-homenet-naming-architecture-dhc-options-02 (work in progress), July 2014.
- [RFC1033] Lottor, M., "Domain administrators operations guide", RFC 1033, November 1987.

Appendix A. Document Change Log

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

-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
Orange
38 rue du General Leclerc
92794 Issy-les-Moulineaux Cedex 9
France

Phone: +33 1 45 29 60 52
Email: daniel.migault@orange.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: January 1, 2015

P. Pfister
B. Paterson
Cisco Systems
J. Arkko
Ericsson
June 30, 2014

Prefix and Address Assignment in a Home Network
draft-pfister-homenet-prefix-assignment-02

Abstract

This memo describes a home network prefix and address assignment algorithm running on top of any 'flooding protocol' that fulfills the specified requirements. It is expected that home border routers are allocated one or multiple IPv6 prefixes through DHCPv6 Prefix Delegation (PD) or that prefixes are made available through other means. An IPv4 address can also be assigned and private addresses be used with NAT to provide IPv4 connectivity. In both cases, provided prefixes need to be efficiently divided among the multiple links, and routers need to obtain addresses. This document describes a distributed algorithm for IPv4 and IPv6 prefixes division, assignment and router's address assignment, and specifies how hosts can be given addresses and configuration options using DHCP or SLAAC.

Status of This Memo

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

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

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

This Internet-Draft will expire on January 1, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	3
2. Requirements language	4
3. Prefix and Address Assignment Algorithms' Outline	4
4. Router Behavior	5
4.1. Data structures	6
4.2. Routers' Interfaces	7
4.3. Obtaining a Delegated Prefix	7
4.4. Network Leader	8
4.5. Designated Router	8
4.5.1. Sending Router Advertisement	9
4.5.2. DHCP Server Operations	9
4.6. Applying an Assignment on an Interface	9
4.7. DNS Support	10
5. Flooding Protocol Requirements	10
5.1. Router ID	11
5.2. Propagation Delay	11
5.3. Flooding Assigned Prefixes	11
5.4. Flooding Delegated Prefixes	12
5.5. Flooding Routers' Address Assignments	12
6. Prefix Assignment Algorithm	12
6.1. When to execute the Prefix Assignment Algorithm	13
6.2. Assignment Precedence	13
6.3. Testing Assignment's validity	14
6.4. Testing Assignment's availability	14
6.5. Accepting an Assigned Prefix	14
6.6. Making a New Assignment	14
6.7. Using Authoritative Prefix Assignments	16
6.8. Choosing the Assignment's Priority	16
6.9. Prefix Assignment Algorithm steps	17
6.10. Downstream DHCPv6 Prefix Delegation support	18
7. Address Assignment Algorithm	18
7.1. Router's address pools	19
7.2. Address Assignment Algorithm	19
8. Hysteresis Principle	20
8.1. Prefix and Address assignments	20
8.2. Delegated Prefixes	20

9. ULA and IPv4 Prefixes Generation	21
9.1. ULA Prefix Generation	21
9.2. IPv4 Private Prefix Generation	21
10. Manageability Considerations	22
11. Documents Constants	22
12. Security Considerations	22
13. References	23
13.1. Normative References	23
13.2. Informative References	24
Appendix A. Scarcity Avoidance Mechanism	25
A.1. Prefix Wasts Avoidance	25
A.2. Increasing Assigned Prefix Length	27
A.3. Foreseeing Prefixes Exhaustion	27
A.4. Cutting an Existing Assignment	28
Appendix B. Acknowledgments	28
Authors' Addresses	28

1. Introduction

This memo describes a fully distributed prefix and address assignment algorithm for home networks, running on top of any 'flooding protocol' that fulfills the specified requirements. It is expected that home border routers are allocated one or multiple IPv6 prefixes through DHCPv6 Prefix Delegation (PD) [RFC3633] or that prefixes are made available through other means. When an IPv4 address is assigned, a home private IPv4 prefix may be used with NAT to provide IPv4 connectivity to the whole home, as well as Unique Local Address prefixes [RFC4193] may be used in order to provide internal connectivity whenever global IPv6 connectivity is not available.

Obtained IPv6 or IPv4 prefixes need to be efficiently divided among the multiple links. For the purposes of this document, we refer to this process as prefix assignment. This memo describes an algorithm for such prefix division, assignment and router's address assignment, as well as the way hosts can be given addresses and configuration options using DHCPv4 [RFC2131], DHCPv6 [RFC3315] or SLAAC [RFC4862]. In the rest of this document DHCP refers to both DHCPv4 and DHCPv6.

Although this document recommends the use of 64 bits long prefixes, the algorithm do not require routers to assign prefixes of particular lengths. When a delegated prefix is too small considered the number of links in the home network, higher priority links may be privileged or smaller prefixes can be assigned in order to avoid prefix scarcity.

The rest of this memo is organized as follows. Section 2 defines the usual keywords, Section 3 outlines the algorithms functioning and features, Section 4 describes how a home router behaves when running

the prefix and address assignment algorithm. Requirements for the underlying flooding protocol are detailed in Section 5. The prefix assignment algorithm is detailed in Section 6 and Section 7 focuses on the address assignment algorithm. Section 8 explains the hysteresis principles applied to both prefix and address assignments, Section 9 specifies the procedures for automatic generation of ULA and IPv4 prefixes, Section 10 explains what administrative interfaces are useful for advanced users that wish to manually interact with the mechanisms, Section 11 gives values for the constants used in this document, Section 12 discusses the security aspects and finally, Appendix A provides implementation guidelines for the optional scarcity avoidance mechanism.

The Prefix Assignment Algorithm was first detailed in [I-D.arkko-homenet-prefix-assignment]. This document is a continuation and generalization of that draft to any underlying flooding protocol. It also adds support for arbitrary prefix length, IPv4, scarcity avoidance mechanism or manual configuration.

2. Requirements language

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

3. Prefix and Address Assignment Algorithms' Outline

Given one or multiple prefixes for the entire network, each prefix is subdivided by the prefix assignment algorithm so that every link is given one assignment per available prefix. Assignments are advertised through the whole network using the underlying flooding protocol, collisions are detected and valid assignments are chosen and applied on every link. Once a prefix is applied, hosts and routers may be given addresses. In summary, the algorithm works in four steps:

1. The home is given IPv6 or IPv4 prefixes called Delegated Prefixes (DPs).
2. Each link is provided an Assigned Prefix (AP) from each available Delegated Prefix.
3. Routers internally check for AP's validity and select Chosen Prefixes (CPs).
4. Once a link is given an assignment, routers may get addresses from specified address pools and hosts may be configured using SLAAC or by the per-link elected DHCP server.

This algorithm, which intends to fulfill requirements specified in [I-D.ietf-homenet-arch], has the following features:

- o Each delegated prefix is effectively divided so that each link is assigned a reasonable part. If the delegated prefix is too small given the size of the network, prefixes of arbitrary lengths may be used.
- o The algorithm is completely distributed. Routers may join or leave and DPs may be added or removed at any time.
- o IPv4 connectivity is provided when a home router acquires an IPv4 address and default route from an external source. In this case a private IPv4 delegated prefix is generated and prefixes are assigned similarly to IPv6.
- o The network may spontaneously generate and use a Unique Local Address (ULA) prefix.
- o Assignments are stable across reboots and some network changes (e.g. adding or removing routers).
- o DHCP options like DNS servers, prefix colors [I-D.bhandari-dhc-class-based-prefix], or any upcoming options may be attached to each prefix and may be relayed down to the host when it is given addresses.
- o The user can manually assign prefixes to links. Such assignments will take precedence over automatically assigned prefixes.
- o Assignments and interfaces can be given priorities. When a delegated prefix is too small, such values may be used to prioritize prefix assignment to certain links.

4. Router Behavior

All home routers participating in the prefix assignment algorithm MUST fulfill the requirements defined in this document and use a common flooding protocol and routing protocol. Classic CPE routers [RFC7084] are supported as downstream routers and downstream DHCPv6-PD enabled routers are supported as both downstream and uplink routers, but problems may occur when such router is connected to the home network on both WAN and LAN side. In the later case, finer external interface detection algorithm or static configuration can be used to solve the issue, but these are out of the scope of this document.

4.1. Data structures

Each router MUST maintain a list of all the Delegated Prefixes. These prefixes may be locally generated, as described in Section 4.3, or come from other routers as described in Section 5.4.

Each router MUST maintain a list of all the Assigned Prefixes advertised by other routers. Each AP is learnt through the mechanisms described in Section 5.3 and is defined as a tuple of:

Prefix: The assigned prefix.

Router ID: The identifier of the advertising router.

Link ID: If the assignment is made on a connected link, an interface identifier of the interface connected to that link.

Authoritative bit: A boolean that tells whether the assignment comes from a network authority (DHCPv6 PD, manual configuration, etc...).

Assignment's Priority: A value between PRIORITY_MIN and PRIORITY_MAX, specifying the assignment's priority.

The AP list is the result of the information provided by the flooding protocol, as specified in Section 5.3.

The router MUST maintain a list of all prefixes currently chosen to be applied on connected links. They are Chosen Prefixes (CPs) and described by a tuple of:

Prefix: The assigned prefix.

Link ID: An interface identifier of the interface connected to the link on which the assignment is made.

Authoritative bit: A boolean that tells whether the assignment comes from a network authority (DHCPv6 PD, manual configuration, etc...).

Assignment's Priority: A value between PRIORITY_MIN and PRIORITY_MAX, specifying the assignment's priority.

Advertised: Whether that assignment is being advertised by the flooding protocol (see Section 5.3).

Applied: Whether that assignment is applied on link's configuration (see Section 4.6).

Chosen Prefixes that are marked as 'Advertised' are distributed to other routers using the flooding protocol and are therefore considered as Assigned Prefixes by other routers. The goal of the Prefix Assignment Algorithm is to ensure that all routers have a consistent view of Assigned Prefixes on each link.

The router MUST maintain a database of its own address assignments, and address assignments made by other routers on connected links as learnt through means described in Section 5.5.

4.2. Routers' Interfaces

Each interface MUST either be considered as internal or external. Prefixes and addresses are only assigned to internal interfaces. The criteria to make this distinction are out of the scope of this document.

If an internal interface becomes external, all prefixes and addresses assigned on the considered interface MUST be deleted and no longer announced, and the prefix assignment algorithm MUST be run.

If an external interface becomes internal, the prefix assignment algorithm MUST be run (see Section 6.1).

Whenever two or more interfaces are connected to the same link, all but one of them SHOULD be ignored by the prefix assignment algorithm. A mechanism to detect such situation SHOULD be provided by the flooding algorithm.

4.3. Obtaining a Delegated Prefix

A Delegated Prefix can be obtained or generated through different means:

- o It can be dynamically delegated, for instance using DHCPv6 PD.
- o It can be created statically, specified in router's configuration.
- o A ULA prefix may be spontaneously generated as defined in Section 9.1.
- o An IPv4 private prefix may be spontaneously generated as defined in Section 9.2.

DHCP options MAY be attached to a delegated prefix by the router that either generated the prefix or received it through DHCPv6 PD. IPv6 delegated prefix options MUST be encoded as DHCPv6 options. IPv4 delegated prefix options MUST be encoded as DHCPv4 options.

As DHCP options are numerous and new ones may be defined, specifying routers' behavior regarding each option is out of the scope of this document. In order to avoid misconfiguration, routers must follow the two following general rules:

- o A router MUST NOT advertise a prefix obtained through DHCPv6 PD if it doesn't understand the all of the provided options.
- o A router MUST NOT make or accept any assignment associated to a delegated prefix if it doesn't understand the all of the DHCP options advertised with the delegated prefix.

The mif working group may provide useful inputs concerning the way the home network should handle different prefixes associated with heterogeneous uplinks.

4.4. Network Leader

A router considers itself as the Network Leader if and only if its router ID is greater than all other router IDs in received Prefix Assignments and Delegated Prefixes.

4.5. Designated Router

On a link where custom host configuration must be provided, or whenever SLAAC cannot be used, a DHCP server must be elected. That router is called designated router and is dynamically chosen by the prefix assignment algorithm.

A router MUST consider itself designated router on a given link if either one of the following conditions holds:

- o The link's Assigned Prefixes list is empty. i.e. no other router is advertising assignments on the considered link. And, if such information is provided by the flooding protocol, the router has the highest id on the link.
- o Considering all APs and advertised CPs on the given link, the router is advertising the one with:
 1. The lowest authoritative bit.
 2. In case of tie, the lowest priority.
 3. In case of tie, the highest router ID.

Note: That particular order (inverted compared to assignments' priority) is motivated by the few cases where a router may

override an existing assignment by advertising an assignment of higher priority. In such a case, the designated router should remain the same.

Example: A new router is powered on and connected to another router that was already there (doing DHCP). It sees the assigned prefix for their common link, but also has, in its own configuration, an authoritative assignment for the link. It starts advertising the authoritative assignment, which causes the second router to remove its previous assignment. Thanks to the inverted order, the DHCP server will remain the same.

4.5.1. Sending Router Advertisement

On a given link, the designated router **MUST** send router advertisements including Prefix Information Options for all the Chosen Prefixes associated to that link. SLAAC **SHOULD** be enabled when possible, unless the configuration states otherwise. The valid and preferred lifetimes **MUST** be set to values lower or equal to the associated Delegated Prefix's valid and preferred lifetimes.

4.5.2. DHCP Server Operations

On a given link, whenever SLAAC can't be used for all assignments, or DHCP configuration options must be provided to hosts, the designated router **MUST** act as a DHCP server and serve addresses on the given link. A router **MUST** stop behaving as a DHCP server whenever it is not the link's designated router anymore.

Routers's addresses pool, specified in Section 7, **MUST** be excluded from DHCP hosts pools.

The valid and preferred lifetimes **MUST** be set to values lower or equal to the associated Delegated Prefix's valid and preferred lifetimes.

4.6. Applying an Assignment on an Interface

Once a Chosen Prefix is created, a router first waits some time in order to detect possible collisions (Section 8). Afterwards and if no collision is detected, the prefix is applied as follows:

- o The router updates its interface configuration so that the prefix is assigned to the considered link.
- o The router updates the routing protocol configuration so that it starts advertising the prefix. Depending on the implementation,

this step may not be needed as the routing protocol directly gets its configuration information from the interfaces configuration.

- o If necessary, the router starts selecting an address for itself as defined in Section 7.
- o If the router is the designated router on the considered link, it starts sending the Prefix Information Option with the considered prefix, as specified in Section 4.5.1.
- o If the router is the designated router on the considered link and if the prefix requires DHCP configuration, it starts behaving as a DHCP server, as defined in Section 4.5.2, for the considered assigned prefix.

When a prefix assignment is removed, the previous steps MUST be undone. The router MUST also deprecate the prefix, if it had been advertised in Router Advertisements on an interface. The prefix is deprecated by sending Router Advertisements with the PIO's preferred lifetime set to 0 [RFC4861]. Hosts that support DHCP reconfigure extension ([RFC3203], [RFC3315]) and that have been given leases MUST be reconfigured as well.

4.7. DNS Support

DHCP options attached to each delegated prefixes and propagated through the flooding protocol SHOULD contain the DHCP DNS options provided by the ISP (when provided).

Whenever the router knows which DNS server to use, or is acting as a DNS relay, it SHOULD include DNS DHCP options ([RFC3646]) within host's configuration messages and include the Router Advertisement DNS options ([RFC6106]) when sending RAs.

DNS server selection in multi-homed networks is a complex issue that this document doesn't intend to solve. One should look at IETF's mif working-group documents in order to obtain guidelines concerning DNS server selection. It is RECOMMENDED that designated routers turns on a local DNS relay that fetches information from provided DNS servers.

5. Flooding Protocol Requirements

In this document, the Flooding Protocol (FP) refers to a protocol enabling information propagation to the whole network. It was not specified in order to allow the working group to independently decide which routing protocol, configuration protocol, and prefix assignment method to use within the home network. Routing protocol, like OSPFv3 [RFC5340] (With its autoconf extension

[I-D.ietf-ospf-ospfv3-autoconfig]) or IS-IS [RFC5308], could be extended in order to fulfill the requirements. An independent protocol, for instance HNCP [I-D.ietf-homenet-hncp], could be used as well.

The specified algorithm can use any protocol that fulfills the requirements specified in this section.

5.1. Router ID

The FP MUST provide a router ID. ID collisions within the network MUST be rare and any conflicts MUST be resolved by the flooding protocol. When the router ID is changed, the FP MUST immediately provide the new ID to the Prefix Assignment Algorithm, which will in turn be run again, without requiring the current state to be flushed.

In the absence of collisions, the router ID MUST NOT be changed, and it SHOULD be stable across reboots, power cycling and router software updates.

5.2. Propagation Delay

The FP MUST provide an approximate upper bound of the time it takes for an update to be propagated to the whole network. This value is referred to as the FLOODING_DELAY. The algorithm ensures that, as long as the upper bound is respected, two identical prefixes will never be applied to different links, and two different prefixes will never be applied to the same link. The algorithm and the network will recover when the upper bound is exceeded, but collisions may appear in the routing protocol and errors may be propagated to upper layers.

If the FP supports link-local flooding, which is used for router's address assignments, it SHOULD provide an approximate upper bound of the time it takes for an update to be propagated to a single link. This value is referred to as the FLOODING_DELAY_LL. If link-local flooding is not available, or the value is not provided, the assignment algorithm MUST use the FLOODING_DELAY value instead.

5.3. Flooding Assigned Prefixes

The FP MUST provide a way to flood Chosen Prefixes marked as advertised and retrieve prefixes assigned by other routers (APs). Retrieved APs MUST contain all the information specified in Section 4.1.

5.4. Flooding Delegated Prefixes

The FP must provide a way to flood Delegated Prefixes and retrieve prefixes delegated to other routers. Retrieved entries must contain the following information.

Prefix: The delegated prefix.

Router ID: The router ID of the router that is advertising the delegated prefix.

Valid until: A time value, in absolute local time, specifying the prefix validity time.

Preferred until: A time value, in absolute local time, specifying the prefix preferred time.

DHCP information: DHCP options attached to the delegated prefix.

The FP MUST make sure time values are consistent throughout the network (i.e. differences are small compared to Delegated Prefixes lifetimes). If no time synchronization protocol is used, the FP MUST keep track of prefix age across the network and within its database.

5.5. Flooding Routers' Address Assignments

Routers addresses are dynamically allocated, picked from a defined pool, and collisions must be detected using the FP. The FP MUST provide a way to flood routers' addresses. The flooding scope of those values SHOULD be link-local, but as addresses are unique within the home network, this is not mandatory. For each address assignment, the FP SHOULD provide the identifier of the interface connected to the link the address assignment was advertised on.

6. Prefix Assignment Algorithm

The Prefix Assignment Algorithm is a distributed algorithm that assigns one prefix from each available Delegated Prefix on every link that is considered to be internal by at least one connected router. The algorithm itself does not distinguish between global IPv6, ULA or IPv4 prefixes. IPv4 prefixes are encoded as their IPv4-mapped IPv6 form, as defined in [RFC4291] (i.e. ::ffff:A.B.C.D/X with X >= 96).

When the Prefix Assignment Algorithm is executed, combinations of Delegated Prefixes and internal interfaces are being considered. For the purpose of this discussion, the Delegated Prefix will be referred to as the current Delegated Prefix, and the interface will be referred to as the current Interface. If a delegated prefix is

included inside another delegated prefix, it is ignored. This rule intends to ignore prefixes delegated from non-Homenet routers that previously obtained their larger prefix from one of Homenet's routers.

The algorithm is specified here for the sake of clarity. It can be optimized in some cases. For instance Prefix Assignment deletion might not need to trigger algorithm's execution if all internal interfaces already have assignments associated to the same Delegated Prefix. Similarly, when an ignored Delegated Prefix is deleted, it is not necessary to run the algorithm. An implementation may work differently than specified here as long as the resulting behavior is identical to the behavior a router implementing this exact algorithm would have.

6.1. When to execute the Prefix Assignment Algorithm

The algorithm **MUST** be run whenever one of the following event occurs:

- o A Delegated Prefix is created or deleted (A DP must be deleted when its lifetime is exceeded).
- o A Prefix Assignment is created, deleted or modified.
- o The router ID is modified.
- o An external link becomes internal, or an internal link becomes external.

It is not required that the algorithm is synchronously run each time such an event occurs. But the delay between the event and the algorithm execution **MUST** be small compared to `FLOODING_DELAY`.

6.2. Assignment Precedence

An assignment is said to take precedence over another assignment when:

- o The authoritative bit value is higher.
- o In case of tie, the priority value is higher.
- o In case of tie, the advertising router's ID is higher.

6.3. Testing Assignment's validity

An Assigned Prefix or a Chosen Prefix is said to be valid if all the following conditions are met:

1. Its prefix is included in an advertised Delegated Prefix.
2. The prefix is not included or does not include any other Assigned Prefix with a higher precedence.
3. No other assignment which prefix is included in the same Delegated Prefix, and with a higher precedence, is being advertised on the same link.

6.4. Testing Assignment's availability

A prefix is said to be available if it does not overlap with any other assignment by any other router in the network.

6.5. Accepting an Assigned Prefix

An AP is said to be accepted when the AP is currently being advertised by a different router on a directly connected link, and will be used by the accepting router as a new Chosen Prefix. When a router accepts a neighbor's assignment, it starts a timer as specified in Section 8. A new CP is created from the AP, with:

- o The same prefix.
- o The same link ID.
- o The authoritative bit set to false.
- o The same priority.
- o The advertised bit value set as specified by the algorithm.
- o The applied bit is unset. It is set when the timer elapsed if the entry still exists.

6.6. Making a New Assignment

When the algorithm decides to make a new assignment, it first needs to specify the desired size of the assigned prefix. Although that choice is completely implementation specific, prefixes of size 64 are RECOMMENDED. The following table MAY be used as default values, where X is the length of the delegated prefix.

If $X < 64$: Prefix length = 64

If $X \geq 64$ and $X < 104$: Prefix length = $X + 16$ (up to 2^{16} links)

If $X \geq 104$ and $X < 112$: Prefix length = 120 (2^8 addresses per link and more than 2^8 links)

If $X \geq 112$ and $X \leq 128$: Prefix length = $120 + (X - 112)/2$ (Link Vs Addresses tradeoff)

When the algorithm decides to make a new assignment, it SHOULD first check its stable storage for an available assignment that was previously applied on the current interface and is part of the current delegated prefix. If no available assignment can be found that way, the new prefix MUST be randomly selected among a subset of available prefixes (if possible, large enough to avoid collisions). Hardware specific identifiers may be used to seed a pseudo-random generator.

If no available prefix is found, the assignment fails.

The algorithm leaves much room for implementation specific policies. For instance, static prefixes may be configured as specified in Section 10. If implemented, the router MAY also decide to execute the Prefix Scarcity Avoidance mechanisms, as proposed in Appendix A.

If an available prefix is found, a new assignment is made and a new Chosen Prefix entry is created.

- o The prefix value is set to the chosen prefix.
- o The link ID is the ID of the link on which the assignment is made.
- o The authoritative bit is set to false.
- o The priority is set to a value between `PRIORITY_AUTO_MIN` and `PRIORITY_AUTO_MAX` (Section 6.8).
- o The advertised bit is set.
- o The applied bit is unset. It is set when the timer elapsed if the entry still exists.

A new assignment is always marked as advertised when created and therefore immediately provided to the flooding protocol.

6.7. Using Authoritative Prefix Assignments

When some authority (Delegating router, system admin, etc...) wants to manually enforce some behavior, it may ask some router to make an Authoritative Prefix Assignment. Such assignments have their Authoritative bit set, CAN NOT be overridden, and will appear in other router's database as Assigned Prefixes with the Authoritative bit set.

There are two kinds of Authoritative Prefix Assignments.

- o When an authority wants to assign some particular prefix to some interface, an Authoritative Prefix Assignment CAN be created and consists in a Chosen Prefix which have its Authoritative bit set and which is advertised. Just like normal assignments, it MUST NOT be applied before the delay specified in Section 8 elapsed.
- o When an authority wants to prevent some prefix from being used, an Authoritative Assignment CAN be advertised. Such assignments MUST NOT be applied and MUST be advertised through the flooding protocol as assigned to either no-interface, or a fake interface (Depending on the flooding protocol's capabilities).

When a delegated prefix is obtained through DHCPv6 PD with a non-empty excluded prefix, as specified in [RFC6603], an Authoritative Prefix Assignment MUST be created with the excluded prefix.

Note: If the router doesn't understand the excluded prefix DHCPv6 option, the delegated prefix is ignored, as specified in Section 4.3.

6.8. Choosing the Assignment's Priority

When either a new Prefix Assignment is made, or an Authoritative Prefix Assignment is created, the creating router needs to choose which priority value to use. The assignment priority is kept by the designated router when it starts advertising the assignment, and is useful when not enough prefixes are available.

- o PRIORITY_DEFAULT SHOULD be used as default.
- o Other values between PRIORITY_AUTO_MIN and PRIORITY_AUTO_MAX MAY be dynamically chosen by the implementation.
- o Other values between PRIORITY_AUTHORITY_MIN and PRIORITY_AUTHORITY_MAX MUST NOT be used if not specified by an authority (by static or dynamic configuration).

- o Other values are reserved.

6.9. Prefix Assignment Algorithm steps

At the beginning of the algorithm, all assignments that do not have their Authoritative bit set are marked as 'invalid', and the router computes for each connected link whether it is the designated router, as specified in Section 4.5.

The following steps are then executed for every combination of delegated prefixes and interfaces.

- o If the current interface is external, ignore that interface.
- o If the Delegated Prefix is strictly included in another Delegated Prefix, ignore that delegated prefix.
- o If the Delegated Prefix is equal to another Delegated Prefix, advertised by some router with an higher router ID than the considered delegated prefix, ignore that delegated prefix.
- o Look for a valid Assigned Prefix, advertised by another router on the current interface and included in the current Delegated Prefix.
- o Look for a Chosen Prefix associated to the current interface and included in the current Delegated Prefix.
- o There are four possibilities at this stage.
 1. If no AP is found, and no CP is found, a new assignment MUST be made if and only if the router considers itself as the designated router. See Section 6.6.
 2. If an AP is found, and no CP is found, the AP MUST be accepted. The new CP's advertised bit MUST be set if and only if the router considers itself as the designated router.
 3. If no AP is found, and a CP is found, the router MUST check if the CP's assignment is valid. If it is, the local assignment is marked as valid and advertised. If it isn't, it is destroyed and the algorithm applies case 1.
 4. If both an AP and a CP are found, the router must check if the prefixes are the same. If they are different and if the CP's Authoritative bit is not set, the CP MUST be deleted and the algorithm applies case 2. If the prefixes are the same, the CP must be updated with the AP's priority value, marked as

valid, and advertised if and only if the router considers itself as designated on the link.

In the end all the assignments that are marked as invalid are deleted.

6.10. Downstream DHCPv6 Prefix Delegation support

If some host or non-Homenet router asks for Delegated Prefixes, a router MAY assign a set of prefixes and give them to the client. Such assignments MUST be advertised as either not assigned on any link, or assigned on a stub virtual link connected to the router, depending on the Flooding Protocol capabilities. By default assignments priorities MUST be between `PRIORITY_AUTO_MIN` and `PRIORITY_AUTO_MAX`, SHOULD be lower than `PRIORITY_DEFAULT`, and the authoritative bit MUST not be set. Whenever such an assignment becomes invalid, DHCPv6 Reconfigure SHOULD be used in order to remove the prefix from DHCPv6 DP client's lease. If DHCPv6 Reconfigure is not supported, leases lifetimes SHOULD be significantly small.

Provided DPs' valid and preferred lifetimes MUST be lower or equal to their associated Delegated Prefix's lifetimes, and associated DHCPv6 data SHOULD be provided to the DHCPv6 PD client.

By default, an assigned prefix SHOULD NOT be provided to a DHCPv6 PD client before the apply timeout has elapsed. But in order to allow faster response delay, a lease MAY first be provided with a lifetime of `2*FLOODING_DELAY` seconds, even if the private assignments' apply timeout has not elapsed yet.

7. Address Assignment Algorithm

IPv6 routers always get at least one link-local address per link. Routing protocols and link DHCP servers are able to run with these addresses. In some cases though, a router may need to take one or multiple addresses among one or multiple available Delegated Prefixes. For example:

- o The router needs connectivity to the internet (For management, NTP synchronization, etc...).
- o The router needs connectivity within the home network (For management, DNS communications, etc...).
- o IPv4 addresses are needed (DHCPv4, v4 link-local connectivity, etc...).

When possible, SLAAC MUST be used. In other cases a different mechanism is necessary for routers to get addresses. This document proposes an Address Assignment Algorithm that extends the Prefix Assignment Algorithm and works as follows. Each prefix assignment is associated with a fixed address pool, reserved for router's addresses assignment. The address pool is a prefix which value is deterministically function of the assigned prefix. A router CAN, at any time, decide to assign itself an address from any of its Chosen Prefixes. Just like prefix assignments, address assignments are advertised to other routers and collisions are detected. Routers MUST keep track of Address Assignments made by other routers on connected links by using information provided by the flooding algorithm, as defined in Section 5.5.

7.1. Router's address pools

Given an assigned prefix A/X (where all A's latest '128 - X'th bits are set to 0), the routers reserved address pool is defined as follows:

If $X \leq 64$: SLAAC MUST be used

If $X > 64$ and $X \leq 110$: The pool is A/112 (2^{16} addresses)

If $X \geq 110$ and $X \leq 126$: The pool is A/(X + 2) (One quarter of the available addresses)

If $X \geq 126$: Only the designated router CAN use A/128. Other routers MUST NOT get an address.

In the case of IPv4 prefixes, the network address (first address of the address pool) MUST not be used.

7.2. Address Assignment Algorithm

In this section, we say an address assignment is made by some router when it intends to use, or is using the address specified by this assignment. An assignment, made by some router, MUST be advertised on the link on which the assignment is made. Similarly, an address assignment is said to be applied when the address is pushed to the router's interface configuration. It is unapplied otherwise.

Routers MUST store applied address assignments in their stable storage and reuse the same addresses whenever possible. At least the five previously applied addresses SHOULD be stored for each interface.

For a given prefix assignment, an address is said to be available if it is within the router's address pool associated to the prefix assignment, and it is not being advertised by any other router. If the flooding protocol provides interface identifier in the address assignments, looking for collisions on considered link is enough.

A new address assignment MUST be chosen randomly among available addresses. An address assignment MUST NOT be applied when one of the following condition is true.

- o The associated Chosen Prefix is not applied.
- o The timer specified in Section 8 has not elapsed yet.

An address assignment must be deleted whenever one of the following condition becomes true.

- o The associated Chosen Prefix is deleted or moved to another link.
- o Some other router with a higher router ID is advertising the same address on the same link.

8. Hysteresis Principle

8.1. Prefix and Address assignments

When the flooding protocol is started, the router MUST wait FLOODING_DELAY before executing the prefix assignment algorithm for the first time.

Prefix and address assignment algorithms are distributed. Collisions may occur, but network configuration, routing protocols or upper layers should not suffer from these collisions. For this reason, all assignments that could imply collisions are not immediately applied.

- o A router MUST NOT apply a Chosen Prefix before it has waited $2 * \text{FLOODING_DELAY}$. If the entry is valid during the whole waiting time, it MUST be applied to the link it is assigned.
- o A router MUST NOT apply an Assigned Address before it has waited $2 * \text{FLOODING_DELAY_LL}$. If the assignment is valid during the whole waiting time, it MUST be applied to the interface it is assigned.

8.2. Delegated Prefixes

When a router stops advertising a Delegated Prefix, it MUST first deprecate that Delegated Prefix by advertising it for

DP_DEPRECATED_FACTOR*FLOODING_DELAY seconds with zero valid and preferred lifetimes.

When a router receives a deprecated Delegated Prefix advertisement, it must remove the Delegated Prefix from its Delegated Prefixes list.

When a router stops receiving a Delegated Prefix from the Flooding Protocol, it SHOULD keep using that delegating prefix up to a period of min(remaining lifetime, DP_KEEP_ALIVE_TIME) seconds.

9. ULA and IPv4 Prefixes Generation

Although DHCPv6 PD and static configuration are regular means of obtaining IPv6 prefixes, routers MAY, in some cases, autonomously decide to generate a delegated prefix. In this section are specified when and how IPv6 ULA prefixes and IPv4 private prefixes may be autonomously generated.

9.1. ULA Prefix Generation

A router MAY generate a ULA prefix when the two following conditions are met.

- o It is the Network Leader (Section 4.4).
- o No other ULA delegated prefix is advertised by any other router.

A router MUST stop advertising a spontaneously generated ULA prefix whenever another router is advertising a ULA delegated prefix.

The most recently used ULA prefix SHOULD be stored in stable storage by all routers and reused whenever choosing a new ULA delegated prefix. If no ULA prefix can be found in stable storage, it MUST be randomly generated, or generated from hardware specific values.

9.2. IPv4 Private Prefix Generation

A router MAY generate an IPv4 prefix when the two following conditions are met.

- o It has an IPv4 address with global connectivity.
- o No other IPv4 delegated prefix is advertised by any other router.

A router MUST stop advertising an IPv4 prefix whenever another router with an higher router ID is advertising an IPv4 Delegated Prefix.

The IPv4 private prefix must be included in one of the private prefixes defined in [RFC1918]. The prefix 10/8 SHOULD be used by default but it SHOULD be configurable. In the case the address provided by the ISP is already a private address, a different private prefix SHOULD be used. For instance, if the ISP is giving the address 10.1.2.3, 10/8 or any sub-prefix included in 10/8 SHOULD NOT be used. (For instance, 172.16/12 or 192.168/16 can be selected).

10. Manageability Considerations

The algorithm leaves much room for implementation specific features. For instance, ULA prefix as well IPv4 prefix generation may be disabled whenever a global IPv6 is made available. This section details a few other possible configuration options.

The implementation MAY allow each internal interface to be configured with a custom priority value. The specified priority SHOULD then be used when creating new assignments on the given interface. If not specified, the default priority SHOULD be used.

The implementation SHOULD allow manual assignments on given links. When specified, and whenever such an assignment is valid, it MUST be advertised as Authoritative Assignments on the given interface.

11. Documents Constants

PRIORITY_MIN	0
PRIORITY_AUTHORITY_MIN	4
PRIORITY_AUTO_MIN	6
PRIORITY_DEFAULT	8
PRIORITY_AUTO_MAX	10
PRIORITY_AUTHORITY_MAX	12
PRIORITY_MAX	15
DP_DEPRECATE_FACTOR	3
DP_KEEP_ALIVE_TIME	60 seconds

12. Security Considerations

Prefix assignment algorithm security entirely relies on flooding protocol security features. The flooding protocol SHOULD therefore check for the authenticity of advertised information. Security modes may be classified in three categories.

1. The flooding protocol is not protected.
2. The flooding protocol's protection is binary: An allowed router may send any type of packets in the name of other routers.

3. All advertised messages are individually signed by the sender.

Whenever a malicious router attacks an unprotected network, or whenever a malicious router is able to authenticate itself to a network as stated in the second case, it may for example:

- o Prevent other routers to get a stable router ID.
- o Prevent other routers from making assignments by claiming the whole available address space.
- o Redirect traffic to some router on the network.

If a malicious router is able to authenticate itself in a network protected as in the third case, most of the previously listed attacks may still be performed, but traffic could only be redirected toward the origination of the attack, and the source of the attack could be identified.

In any case, in order to protect the network, the routing protocol as well as the way hosts are configured also needs to be protected, hence requiring other link (e.g. WPA) or IP layer (e.g. IPSec-Auth [RFC4302] or SeND [RFC3971]) security solutions.

13. References

13.1. Normative References

- [RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.
- [RFC3203] T'Joens, Y., Hublet, C., and P. De Schrijver, "DHCP reconfigure extension", RFC 3203, December 2001.
- [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.

- [RFC3646] Droms, R., "DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3646, December 2003.
- [RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, October 2005.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.
- [RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.
- [RFC6106] Jeong, J., Park, S., Beloeil, L., and S. Madanapalli, "IPv6 Router Advertisement Options for DNS Configuration", RFC 6106, November 2010.
- [RFC6603] Korhonen, J., Savolainen, T., Krishnan, S., and O. Troan, "Prefix Exclude Option for DHCPv6-based Prefix Delegation", RFC 6603, May 2012.

13.2. Informative References

- [RFC3971] Arkko, J., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, March 2005.
- [RFC4302] Kent, S., "IP Authentication Header", RFC 4302, December 2005.
- [RFC5308] Hopps, C., "Routing IPv6 with IS-IS", RFC 5308, October 2008.
- [RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, July 2008.
- [RFC7084] Singh, H., Beebee, W., Donley, C., and B. Stark, "Basic Requirements for IPv6 Customer Edge Routers", RFC 7084, November 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-11 (work in progress), October 2013.

- [I-D.ietf-homenet-hncp]
Stenberg, M. and S. Barth, "Home Networking Control Protocol", draft-ietf-homenet-hncp-00 (work in progress), April 2014.
- [I-D.ietf-ospf-ospfv3-autoconfig]
Lindem, A. and J. Arkko, "OSPFv3 Auto-Configuration", draft-ietf-ospf-ospfv3-autoconfig-06 (work in progress), February 2014.
- [I-D.arkko-homenet-prefix-assignment]
Arkko, J., Lindem, A., and B. Paterson, "Prefix Assignment in a Home Network", draft-arkko-homenet-prefix-assignment-04 (work in progress), May 2013.
- [I-D.bhandari-dhc-class-based-prefix]
Systems, C., Halwasia, G., Gundavelli, S., Deng, H., Thiebaut, L., Korhonen, J., and I. Farrer, "DHCPv6 class based prefix", draft-bhandari-dhc-class-based-prefix-05 (work in progress), July 2013.
- [I-D.chelius-router-autoconf]
Chelius, G., Fleury, E., and L. Toutain, "Using OSPFv3 for IPv6 router autoconfiguration", draft-chelius-router-autoconf-00 (work in progress), June 2002.
- [I-D.dimitri-zospf]
Dimitrelis, A. and A. Williams, "Autoconfiguration of routers using a link state routing protocol", draft-dimitri-zospf-00 (work in progress), October 2002.

Appendix A. Scarcity Avoidance Mechanism

Although an ISP should provide enough addresses, an implementation must carefully manage the provided address space. First, when a new assignment is made, the prefix should be selected amongst a set of prefixes so that prefix waste is minimized. Then, a router may decide to execute procedures intended to avoid prefix scarcity. Different approaches are possible. This section intends to provide guidelines for such procedures. They are optional and are compatible with routers that only support basic requirements defined in this document.

A.1. Prefix Wasts Avoidance

Given a Delegated Prefix, different routers may try to assign prefixes of different lengths. Particularly, a non-homenet downstream router may ask for a delegated prefix of significant size,

as specified in Section 8.2. Some other routers, like sensors, may also require small prefixes. When randomly selected, a few /80s may easily prevent the assignment of bigger prefixes. Small prefixes should therefore be selected in neighboring areas.

For instance, given a delegated prefix 2001::/56 and an assigned prefix 2001::/64, the best prefix choice in order to reduce prefix space waste is 2001:0:0:1::/64. Other choices are then to be taken in 2001:0:0:2::/63, 2001:0:0:4::/62, 2001:0:0:8::/61, etc...

Creating an efficient prefix selection algorithm may be challenging as it needs to fullfill somehow contradictory requirements:

1. The prefix **MUST** be chosen amongst available prefixes, which implies that other routers may interfere with the process.
2. The prefix **MUST** be chosen randomly in a subset of available prefixes. When possible, the subset must be big enough to avoid collisions.
3. The prefix **SHOULD** be selected amongst prefixes that reduces the prefix space waste.
4. The prefix **SHOULD** be selected pseudo-randomly.

The following algorithm offers a satisfying tradeoff. Given a Delegated Prefix and the desired prefix length:

1. Compute the minimal subset of available prefixes included in the Delegated Prefix. In the example given previously, the minimal subset was {2001:0:0:1::/64, 2001:0:0:2::/63, ..., 2001:0:0:80::/57}.
2. Compute the set of prefixes of desired length so that:
 - * It contains exactly RANDOM_SUBSET_SIZE prefixes, or all the available prefixes if there are less than RANDOM_SUBSET_SIZE available prefixes.
 - * Prefixes are picked in the prefixes from the minimal subset of available prefixes which lengths are the highest.
 - * When multiple subsets are possible, privelege lexicographically lowest prefixes.

If RANDOM_SUBSET_SIZE equals 10, the subset would be {2001:0:0:1::/64, 2 /64s in 2001:0:0:2::/63, 4 /64s in 2001:0:0:4::/62, the 3 first /64s in 2001:0:0:8::/61}.

3. First try PSEUDO_RANDOM_TENTATIVE pseudo-random prefixes, computed from the DP, with the given length, based on interface specific hardware values (For instance using values generated like HASH(MAC Address : Counter). The hash function doesn't need to be cryptographic). The first prefix amongst this set that also is in the set computed at step 2 is chosen. If no prefix is found, try next step.
4. Choose a prefix randomly among prefixes in the subset computed at step 2.

This algorithm, defined as a sequence of prefix sets computation, may seem algorithmically complex, but it can be efficiently implemented. The key element in order to do so is the ability to iterate efficiently over all the available prefixes.

RANDOM_SUBSET_SIZE should provide sufficiently low collision probability. A value of 256 should be enough in most cases. PSEUDO_RANDOM_TENTATIVE is purely implementation dependent, but shouldn't be too high as the probability of finding an available prefix that way quickly decreases with the number of used prefixes. A value of 10 should be sufficient.

A.2. Increasing Assigned Prefix Length

When a new assignment can't be created, and if not forbidden by the router's configuration, the router MAY increase the size of the desired prefix. For instance, if an available /64 can't be found, the router may look for a /80. Nevertheless, this implies using DHCPv6 instead of SLAAC, which SHOULD be avoided.

A.3. Foreseeing Prefixes Exhaustion

The previously proposed solution may be useful in some particular cases, but won't work when no more prefixes are available. A router MAY try to detect when default length prefixes are becoming rare. In such a situation, it MAY decide to allocate a longer prefix, part of an available shorter prefix. For instance, if A/64 is available, but there are not many other available /64, the router can try to allocate A/80. If the allocation doesn't raise any collision, this procedure will prevent A/64 from being used by other hosts, hence creating a large set of smaller available prefixes to be used.

Such an allocation is considered dynamic. The Authoritative bit MUST NOT be set and the priority MUST be among values authorized as dynamically chosen in Section 6.8.

When different prefixes lengths are being used, the random prefix selection MUST NOT be uniform among all possibilities. Instead, it SHOULD privilege prefixes contained in bigger prefixes that cannot be allocated. For instance, if 2001::/56 is the DP, and 2001:0:0:0:1::/80 is an assigned prefix, other /80 should be randomly chosen in 2001:0:0:0:1::/64 before being chosen in other /64s.

A.4. Cutting an Existing Assignment

When specifically required by an authority (configuration or DHCP), a router MAY decide to un-assign one of its own assignment, in order to cut it in smaller prefixes, or to send an overriding assignment in order to force the network to stop using a particular prefix. Because such a procedure may imply links reconfiguration, it SHOULD be avoided whenever possible.

Such allocation are considered as required by an authority. The Authoritative bit MAY be set and the priority MUST be among values authorized as specified by an authority in Section 6.8.

As an example, if a router can't find a /64 for a link that, with a high priority, must be given a /64, it chooses a prefix assigned by some other router, to another link, with a lower priority, and creates a new Chosen Prefix with a higher priority. The other router will be forced to remove its own assignment, hence making the new assignment valid.

Appendix B. Acknowledgments

This document is the continuation of the work being done in [I-D.arkko-homenet-prefix-assignment]. The authors would like to thank all the people that participated in the previous document's development as well as the present one. In particular, the authors would like to thank to 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 in this problem space. The authors would also like to point out some past work in this space, such as those in [I-D.chelius-router-autoconf] or [I-D.dimitri-zospf].

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

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 21, 2014

M. Stenberg
June 19, 2014

Auto-Configuration of a Network of Hybrid Unicast/Multicast DNS-Based
Service Discovery Proxy Nodes
draft-stenberg-homenet-dnssd-hybrid-proxy-zeroconf-01

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 December 21, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

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

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

In this document, the key words "MAY", "MUST", "MUST NOT", "OPTIONAL", "RECOMMENDED", "SHOULD", and "SHOULD NOT", 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).(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 larger network, after conflict resolution, identifying which network 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. If 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.

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 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. References

7.1. 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.

7.2. Informative references

- [I-D.ietf-homenet-hnmp] Stenberg, M. and S. Barth, "Home Networking Control Protocol", draft-ietf-homenet-hnmp-00 (work in progress), April 2014.
- [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.

7.3. URIs

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

Appendix A. Example configuration

A.1. Used topology

Let's assume home network that looks like this:

```

      | [0]
    +-----+
    |  CER  |
    +-----+
  [1]/       \ [2]
   /         \
+-----+ +-----+
|  IR1  | - |  IR2  |
+-----+ +-----+
| [3] |   | [4] |

```

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 practise 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

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: November 08, 2014

M. Stenberg
May 07, 2014

Minimalist Port Control Protocol Proxy
draft-stenberg-homenet-minimalist-pcp-proxy-00

Abstract

This document describes a minimalist PCP proxy function needed within the homenet architecture. It is notably a subset of a general PCP proxy.

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 November 08, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	2
2. Requirements language	2
3. Requirements for the design	2
4. The use case for MPP	3
4.1. State required	3
4.2. Difference from 'general' PCP proxy	3
5. Algorithm	3
5.1. Local epoch reset	4
5.2. Client -> Proxy server port (ANNOUNCE)	4
5.3. Client -> Proxy server port -> Server (MAP/PEER)	4
5.4. Server -> Proxy client port -> Client (MAP/PEER)	4
6. References	5
6.1. Normative references	5
6.2. Informative references	5
Appendix A. Draft source	6
Appendix B. Acknowledgements	6
Author's Address	6

1. Introduction

This is mostly discussion fodder; I personally find current PCP proxy defined in [I-D.ietf-pcp-proxy] overcomplex for Homenet needs. So I'm defining Minimalist PCP Proxy (MPP) here instead.

A GPLv2-licensed experimental and probably still incorrect sample implementation of MPP is currently under development at <https://github.com/fingon/minimalist-pcproxy/> [1]. Comments and/or pull requests are welcome.

2. Requirements language

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

3. Requirements for the design

Homenet architecture defined in [I-D.ietf-homenet-arch] allows for multihoming -> multiple PCP servers MUST be supported. Notably, the PCP server choice MUST depend on the source address used by the client.

IPv4 is not yet gone -> dual-stack PCP SHOULD be supported. Proposed homenet prefix assignment algorithm defined in [I-D.pfister-homenet-prefix-assignment] assumes only zero or one upstream IPv4 links, NATted to a single IPv4 prefix.

The amount stored state SHOULD be minimal.

MPP SHOULD also have as simple as possible implementation for both footprint and correctness validation reasons.

4. The use case for MPP

Each first-hop router in a Homenet runs this algorithm. Each router with upstream connectivity additionally runs a real PCP server, but on an IP address that is not provided to any clients (TBD or just some weird port#? We're among consenting routers here after all..). [I-D.ietf-homenet-hnmp] is used to maintain the information about upstream connections for the running MPP instances, and therefore normal PCP server selection is not needed.

4.1. State required

In addition to the local definition of epoch, for each server, following information is stored and updated as needed:

- o Source IP prefix and length to match.
- o Remote IP address of the server.match.
- o Remote epoch tracking (prev_server_time, prev_client_time as per [RFC6887]).

4.2. Difference from 'general' PCP proxy

The MPP defined here is only a subset of what official PCP proxy draft [I-D.ietf-pcp-proxy] covers. However, it also is MUCH simpler to implement and define. Notable limitations include:

- o This scheme cannot be used on PCP proxy nodes that actually perform NAT. In case of firewalling, or forwarding, it should work. This is because original destination address client used to contact the local proxy is reused, to store it for later forwarding the response back to the client. If NAT occurs, this is not possible.
- o MPPs cannot be cascaded.
- o MPPs may be hard to adapt to real server selection in non-Homenet environments (TBD).

5. Algorithm

Next behavior of MPP is described. MPP MUST have both PCP client and PCP server ports open.

5.1. Local epoch reset

On local epoch reset (when MPP is started, or based on detected epoch reset at one of the servers as defined in Section 5.4), MPP SHOULD send unsolicited multicast ANNOUNCEs as specified in [RFC6887].

5.2. Client -> Proxy server port (ANNOUNCE)

Just provide a direct response (given internal interface + local IP), as specified in [RFC6887]. Otherwise, ignore.

5.3. Client -> Proxy server port -> Server (MAP/PEER)

On receipt of a PCP request on an internal interface on the PCP server port, MPP behaves as follows:

- o Check if the source IP address and the PCP client IP Address are the same. If a mismatch is detected, the behavior specified in [RFC6887] must be followed.
- o Check that for the client's source IP address, there exists a PCP server responsible for it within the local configuration. If not, TBD (error, but which one).
- o If THIRD_PARTY is already set, consider the request rejected.
- o If the request is rejected, build an error response and send it back to the PCP client. The error status code is set to NOT_AUTHORIZED.
- o If the request is accepted, adjust it (e.g., adding a THIRD_PARTY Option, updating the PCP client IP Address to the address client used when contacting the proxy) and forward it from local client port with the source address matching the IP address in the adjusted request.

5.4. Server -> Proxy client port -> Client (MAP/PEER)

On receipt of a PCP response on the PCP client port, MPP behaves as follows:

- o Check that source IP matches one of the PCP servers, and that the source port matches PCP server port. If not, silently drop the packet.

- o Check that THIRD_PARTY option is present, and store it for future use. If it is not present, or not in a locally connected prefix, silently drop the packet.
- o Ensure that the per-server epoch is valid per [RFC6887]. If not, reset local epoch.
- o Adjust the epoch in the response to local epoch.
- o Send the request forward to the client, with source address matching the original destination address, the destination address matching the address within the removed THIRD_PARTY option, and from the local server port to the remote client port.

6. References

6.1. Normative references

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC6887] Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", RFC 6887, April 2013.

6.2. Informative references

- [I-D.ietf-pcp-proxy]
Perreault, S., Boucadair, M., Penno, R., Wing, D., and S. Cheshire, "Port Control Protocol (PCP) Proxy Function", draft-ietf-pcp-proxy-05 (work in progress), February 2014.
- [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-11 (work in progress), October 2013.
- [I-D.ietf-homenet-hncp]
Stenberg, M. and S. Barth, "Home Networking Control Protocol", draft-ietf-homenet-hncp-00 (work in progress), April 2014.
- [I-D.pfister-homenet-prefix-assignment]
Pfister, P., Paterson, B., and J. Arkko, "Prefix and Address Assignment in a Home Network", draft-pfister-homenet-prefix-assignment-00 (work in progress), February 2014.

Appendix A. Draft source

As usual, this draft is available at <https://github.com/fingon/ietf-drafts/> [2] 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 B. Acknowledgements

The algorithm text is adapted from draft-ietf-pcp-proxy-04 Section 8. It is unfortunately gone from the more recent iterations.

Author's Address

Markus Stenberg
Helsinki 00930
Finland

Email: markus.stenberg@iki.fi