

Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 4, 2014

M. Boutier
J. Chroboczek
PPS, University of Paris-Diderot
July 3, 2013

Source-specific Routing
draft-boutier-homenet-source-specific-routing-00

Abstract

Source-specific routing is a generalisation of next-hop routing in which the routing decision is made depending on a packet's source address in addition to the destination. We describe the motivation for source-specific routing and our experiences with an experimental extension of the Babel routing protocol that implements source-specific routing.

Status of this Memo

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

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

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

This Internet-Draft will expire on January 4, 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. Introduction	3
2. Source-specific routing	3
2.1. Usage scenarios	3
2.2. Routing tables	4
3. Implementation	6
4. Interoperability issues	7
4.1. Interoperability with next-hop routing	7
4.2. Other forms of specific routing	7
5. Applicability to link-state protocols	8
6. Conclusions	8
7. References	9
Authors' Addresses	9

1. Introduction

The main routing paradigm deployed on the Global Internet is next-hop routing. In next-hop routing, routing decisions are performed per-packet, and consist in examining a packet's destination address only, and mapping it to a next-hop router.

The use of next-hop routing restricts the flexibility of the routing system in two ways. First, since a router only controls the next hop, a route can only be selected by the network if it has a selected route as its suffix, which makes some forms of global optimisation difficult or impossible. Other routing paradigms, such as circuit switching, label switching and source routing, do not have this limitation. (Source-routing, in particular, has been proposed multiple times as a suitable routing paradigm for the Global Internet [CLARK]), but has been forbidden due to claimed security reasons [RFC5095].

Second, the only decision criterion used by a router is the destination address. This implies that two packets with the same destination are routed identically, which is not always desirable. There are other data in the IP header that can be reasonably used for making a routing decision -- the TOS octet, the flow-id, and, of course, the source address.

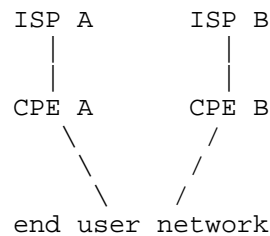
2. Source-specific routing

Source-specific routing is a modest extension of next-hop routing. In source-specific routing, just like in next-hop routing, a router's role is limited to computing a next hop. Unlike in next-hop routing, however, it can use both the destination and the source address in order to perform this computation. In effect, source-specific routing gives a modest amount of routing control to the sending host, which can choose among potentially many source addresses, while leaving routing decisions firmly in the control of the routers.

2.1. Usage scenarios

2.1.1. Simple multihoming

Consider a multihomed network connected to two (or more) providers, for example a home network with two ADSL lines, or one ADSL line and one cellular connection. We assume no cooperation between the two providers, so that there are two edge routers ("CPEs"), one for each provider. We further assume that one or both ISPs might be hostile to multihoming, so that solutions requiring changes to the on-the-wire packet format are not applicable.



Each provider grants the network a range of addresses that can be assigned to nodes. A node can choose to configure with an address from one of the two ranges, or to configure two addresses, one from each range; it will then need to choose, for each packet being sent, an address to use as the source address.

Since providers hopefully implement source address filtering [BCP38], the network must choose the edge router through which to route a packet depending on its source.

2.1.2. Tunnels and VPNs

Tunnels and VPNs are commonly used to establish a network-layer topology that is different from the physical topology, notably for security reasons. In many tunnel or VPN deployments, the end network uses its native default route, and only routes some set of prefixes through the tunnel or VPN.

In some deployments, however, the default route points at the tunnel. If this is done naively, the network stack attempts to route the encapsulated packets through the tunnel itself, which causes the tunnel to break. Many workarounds are possible, the simplest being to point a host route towards the tunnel endpoint through the native interface.

Source-specific routing provides a clean solution to that problem. The native default route is kept unchanged, while a source-specific default route is installed through the tunnel. The source-specific route being more specific than the native default route, packets from the user network are routed through the tunnel, while the encapsulated packets sourced at the edge router follow the native, non-specific route.

2.2. Routing tables

In classical next-hop routing, every router maintains a routing table, a set of pairs (D, NH), where D is a destination prefix and NH the corresponding next-hop router. When a packet with destination address d is routed, an entry (D, NH) such that d is in D is

selected, and the packet is sent to the corresponding NH.

In a source-specific router, the routing table is a set of triples of the form (D, S, NH), where D is a destination prefix, S a source prefix, and NH the next-hop router. When a packet with destination d and source s is routed, an entry (D, S, NH) is selected such that d is in D and s is in S, and the packet is sent to the corresponding NH.

2.2.1. Ambiguity

The two procedures described above omit an important detail: in general, there are multiple routing table entries that match a given packet. A router must therefore choose one among these entries in order to determine a next hop.

In next-hop routing, if two routing table entries overlap, then one is necessarily more specific than the other; the "longest prefix rule" specifies that the most specific applicable routing table entry is chosen. In source-specific routing, this is no longer the case: there might, in general, be multiple applicable entries with none being included in the others.

Consider the following fragment of a routing table:

```
(2001:DB8:0:1::/64, ::/0, A)
(::/0, 2001:DB8:0:2::/64, B)
```

This specifies that all packets with destination in 2001:DB8:0:1::/64 are to be routed through A, while packets with a source in 2001:DB8:0:2::/64 are to be routed through B. A packet with source 2001:DB8:0:2::42 and destination 2001:DB8:0:1::57 matches both rules, although neither is more specific than the other.

We say that a routing table such as the above is ambiguous. Most practical routing tables with source-specific routes turn out to be ambiguous.

2.2.2. Resolving ambiguity

In the presence of ambiguity, routing tables should be considered by destination first; intuitively, "the destination wins". (We are indebted to Fred Baker, who explained that to us.)

Consider the following network topology:

```
::/0 --- A --- B --- C --- 2001:DB8:0:1::/64
```

Suppose that the routing table at B contains a source-specific default route through A and a non-specific route towards 2001:DB8:0:1::/64 through C. The correct behaviour is clearly to send a packet destined to 2001:DB8:0:1::/64 through C -- this is the only choice that has a chance of getting the packet to the right destination.

It is important to note that all routers in the same routing domain must have the exact same behaviour in the presence of ambiguity, lest persistent routing loops occur. Indeed, consider again the example above; if router C implements a "source first" disambiguation behaviour, then it will send B's packets back to B, which in turn will send it back to B, etc.

2.2.3. Disambiguation routes

Ideally, we would like the lower layers of the system (the OS kernel, the line cards, etc.) to implement source-specific routing tables out of the box, with the right disambiguation behaviour already present. In practice, however, we have found that such lower-layer support either doesn't exist, doesn't work, or has a behaviour different from the one desired.

In order to work with the limitations of the lower layers, we introduce disambiguation routes. A disambiguation route is a route that covers the intersection of two ambiguous routes, and therefore specifies the behaviour of packets that match both. Disambiguation routes do not appear on the wire, and in our implementation are not even inserted into the RIB; they are computed and inserted into the FIB on the fly, at route selection time. From the point of view of the routing protocol, disambiguation routes are a lower level implementation detail.

Interestingly enough, we have found that we do not need to maintain a list of disambiguation routes that we have installed: when removing a route from the FIB, the set of disambiguation routes that need to be removed can be computed on the fly, similarly to what happens during route insertion.

3. Implementation

We have implemented a source-specific variant of the Babel routing protocol [BABEL] for the Linux kernel. We first attempted to use the source-specific API provided by Linux; it turns out that this API is specific to IPv6, and only works in a very restricted case, insufficient for our needs.

We have therefore chosen to use the "rule" API, which allows a

routing daemon to use multiple routing tables that are combined using a fairly flexible set of "rules". We use a dynamically allocated set of routing tables, and manage routing rules on the fly. The use of disambiguation routes is essential to obtaining the right behaviour.

4. Interoperability issues

4.1. Interoperability with next-hop routing

In many networks, only some routers will need to perform source-specific routing decisions. For example, in a typical multihomed network the two specific default routes will match in most of the network, and only be distinguished near the edge routers. Our implementation allows running a base version of Babel within most of the network, and only run a source-specific daemon where the specific routes are distinct.

Source-specific routes are encoded within the protocol as a new TLV type, in accordance with the Babel extension mechanism [BABEL-EXT]. This new TLV will be silently ignored by base Babel routers, which will therefore route packets following non-specific routes only.

Hybrid networks consisting of base and source-specific routers do not cause persistent routing loops. However, since non-specific routers do not see source-specific routes, they might drop packets unless they have enough non-specific routes; distributing a non-specific default route throughout the network solves this particular issue in all cases.

Additionally, since non-specific routers do not propagate specific routes, packets may end up routed to the wrong destination unless there are enough specific routers to propagate all the specific routes throughout the network. A simple solution is to ensure that the specific routers form a connected subgraph, which, at worst, can be achieved by using tunnels. Intuitively, such a network consists of a source-specific backbone together with a set of non-specific leaf networks.

4.2. Other forms of specific routing

The technology described in this document is fully general, and applies equally well to other forms of specific routing (say, TOS-specific or flow-id-specific routing). In the presence of multiple forms of specific routing, a natural question to ask is whether they can interoperate in a single routing domain.

In general, such interoperability is possible assuming that the

preference rules of all the implementations are subsets of a single total order on all of the routing criteria; equivalently, there must exist a consistent linearisation of all of the orderings used by the different implementations. Indeed, consider again a simple linear topology:

X --- A --- B --- Y

Suppose that X announces a source-specific default route, while Y announces a flow-id-specific default route. A packet that matches both routes must be treated consistently by A and B, lest a routing loop arise.

A simple (if brutal) way of meeting the linearisation requirement is to require all routers to be specific in one dimension only: to allow a router to perform source-specific routing, flow-id-specific routing, but not both at the same time.

5. Applicability to link-state protocols

While our implementation is an extension of the Babel routing protocol, our work applies equally well to any distance vector routing protocol, such as RIPv2, RIPv6 or EIGRP. The question remains about link-state routing protocols.

The currently deployed link-state protocols (OSPF and IS-IS) are actually hybrid protocols: they divide the network into areas, and perform link-state routing within areas and distance-vector routing within areas. We are therefore confident that our techniques can be used to extend link-state protocols with source-specific inter-area routing (a simplified case of that has been implemented for OSPF [STENBERG]); in OSPF terms, source-specific routes are analogous to Type 5 LSAs.

Whether it is possible to extend the current link-state protocols with support for intra-area source-specific routing, or whether it is desirable to do so, are currently open questions.

6. Conclusions

Source-specific routing is a modest extension to ordinary next-hop routing that makes a number of useful scenarios possible. In this document, we have described the difficulties associated with source-specific routing, and described the solution we adopted within our implementation of source-specific routing within the Babel routing protocol.

We expect our experience to be useful to future implementers of source-specific routing within other routing protocols.

7. References

- [BABEL] Chroboczek, J., "The Babel Routing Protocol", RFC 6126, February 2011.
- [BABEL-EXT] Chroboczek, J., "Extension Mechanism for the Babel Routing Protocol", Internet Draft draft-chroboczek-babel-extension-mechanism-00, June 2013.
- [BAKER] Baker, F., "IPv6 Source/Destination Routing using OSPFv3", Internet Draft draft-baker-ipv6-ospf-dst-src-routing-00, February 2013.
- [BCP38] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, May 2000.
- [CLARK] Reed, D. and D. Clark, "Source Routing for Campus-wide Internet Transport", September 1980.
- [RFC5095] Abley, J., Savola, P., and G. Neville-Neil, "Deprecation of Type 0 Routing Headers in IPv6", RFC 5095, December 2007.
- [STENBERG] Stenberg, M., "Hnet (core) package", 2012, <<https://github.com/fingon/hnet-core>>.
- [TROAN] Troan, O. and L. Colitti, "IPv6 Multihoming with Source Address Dependent Routing (SADR)", Internet Draft draft-troan-homenet-sadr-00.

Authors' Addresses

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

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

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

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

Internet Engineering Task Force
Internet-Draft
Intended status: Experimental
Expires: January 17, 2014

C. Dessez
Cisco Systems
July 16, 2013

Connecting Home Networks via the social network GooglePlus
draft-dessez-homenet-googleplus-interconnect-01

Abstract

This document describes an experimental implementation for connecting home networks via a social network. The social network is used to extend the boundary of a single home network to include other home networks. In this way, access to devices or services within a home can be granted among home networks based on their relation to one another within the social network.

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 17, 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. Introduction	3
1.1. Terminology and abbreviations	3
2. Defining the set of connected homes	4
3. Overall architecture	5
4. Network architecture	6
4.1. Managing the tunnels	6
4.2. Configuring the network	7
5. Sharing services within your set of connected homes	8
6. Security Considerations	9
7. Experimental results	10
8. IANA Considerations	10
9. Acknowledgements	10
10. References	10
10.1. Normative References	10
10.2. Informative References	11
Author's Address	11

1. Introduction

The goal of this experiment is to allow an average home user to extend the boundaries of their home network to other home networks the user trusts. Other home networks may be owned by a single user, or "friends" of the user as defined by a social network. This document describes an overall architecture and specific mechanisms chosen for a working implementation based on the social network Google Plus.

In each home, one router is responsible for interacting with the social network. The home network is represented within the social network as a "Page" which the user owns. The router is given credentials to interact with its representative Page, while the user defines the relationship of the Pages to one another. When a bidirectional relationship between two home network Pages is detected, the information necessary to setup a tunnel is shared by posting it to the social network. An encrypted tunnel is then setup between the homes, and a link established.

IP reachability among linked homes is achieved by insertion and propagation of routes into a routing protocol running within the home network. Services are then advertised among homes as defined in [I-D.cheshire-mdnsext-hybrid] and [I-D.stenberg-homenet-dnssdext-hybrid-proxy-ospf]. Finally, by connecting to a UI hosted by the specific router, the user can define policies for the services permitted to be shared within a given circle defined by the social network.

The mechanisms described in the following sections assume a homenet environment as described in [I-D.ietf-homenet-arch] with with a routing protocol such as that defined in ([I-D.acee-ospf-ospfv3-autoconfig]) as well as the mechanism of prefix assignment defined in [I-D.arkko-homenet-prefix-assignment] .

1.1. Terminology and abbreviations

In this section we define terminology and abbreviations used throughout the text.

- o Homenet: a home network as defined in [I-D.ietf-homenet-arch]
- o Gplus: Google Plus. Google's social network.
- o Gplus router: the router that is responsible for the connection to Google Plus, on which the mechanisms described in this document are hosted.

- o Circle: represents a group of people for which you can define confidentiality and visibility policies in Google Plus.
- o Gplus ID: the unique internal identifier of an entity in Google Plus. It apparently consists in a decimal number on the order of 10^{21} for users and pages accounts, and a 64-bit hexadecimal number for circles.
- o DNS-SD: DNS-Based Service Discovery [RFC6763].
- o ULA: IPv6 Unique Local Addresses [RFC4193].
- o CA: Certificate Authority (as defined in X.509 [RFC3280]).
- o CRT: an X.509 certificate ([RFC3280])
- o CSR: Certificate Signing Request or Certificate Request ([RFC3280]).
- o CPE: Customer Premises Equipment.

2. Defining the set of connected homes

The central idea of this experiment is for the homenet to be represented within the social network in a way that is intuitive to the user. For this to happen, the homenet must be represented in a way such that:

- o the homenet is clearly linked to its owner
- o the user can manage the relationships of the homenet with other homenets linked to other users
- o the network devices in the homenet can retrieve its social topology and setup communication with its related homenets

If social networks were widely used for connecting homenets today, there may be some specific entity that a user could define that would clearly be identified as a home network. This would be available for setting up connections to, based on the users policy and relationship to other users with homenets as part of their profile. As that is not the case today among popular social networks such as Facebook and Google Plus, we looked into what might be the closest fit and decided to use Google Plus pages. Intended mainly for brands and businesses, they are not very different from user accounts on a social point of view (they organize their contacts and what they see by the system of circles). A user may have several pages, and a page may have several

administrators, each of them being able to easily log in as the page while connected to their regular Gplus account.

In this implementation, the home router connects to Gplus to retrieve the topology and communicate with other routers using the Google Pages API. This API uses OAuth 2.0 ([RFC6749]) to allow the user to delegate the management of pages to their Gplus router.

In Gplus, the relationships between people and pages are ruled by the system of circles. One can circle whoever they want in one or more of their circles, without it needing to be accepted by the latter. But in our case, we consider a tunnel must be created only if the relationship is bidirectional, that is only if they have both circled each other in at least one circle. Notice that whereas one cannot know what are the circles of someone else, they know who has circled them, which is enough to know whether a relationship is bidirectional. The Section 5 will explain in details how the visibility policies of DNS-SD services are directly linked to circles.

As stated earlier, the router needs to send messages through Gplus in order to exchange the information necessary to establish and configure the tunnel. This information can be divided into three categories: routing information, cryptographic keys and DNS-SD settings. The routing information and the DNS-SD settings, which we will call Network Settings, are gathered in a post that is regularly updated and visible to everyone in the page's circles. This will be detailed in . As for the posts conveying cryptographic keys, they will be described in Section 4.

3. Overall architecture

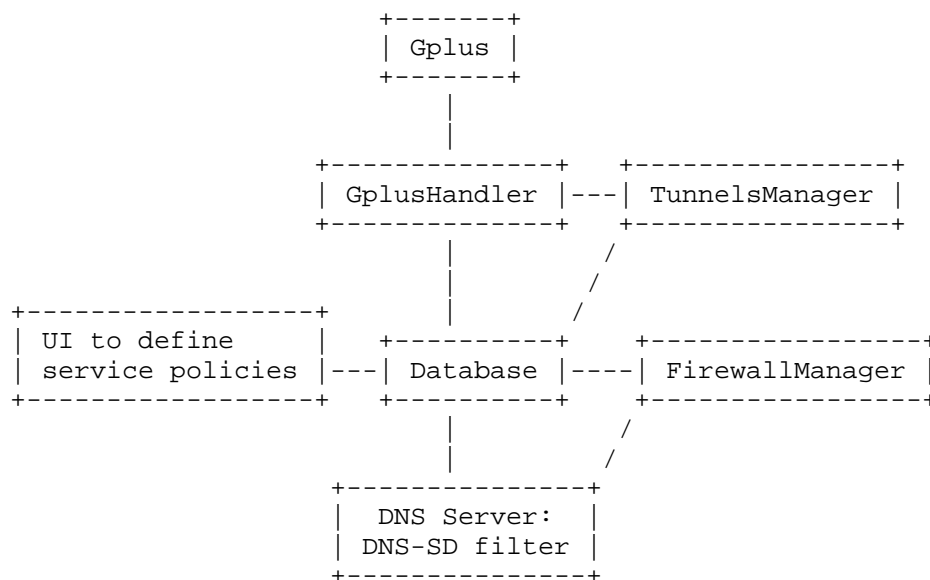
Figure 1 represents the global functional architecture of the implementation and shows the interactions between its different parts.

The interaction with Gplus is handled by a module called GplusHandler. It performs regular polling to update the social topology in the database, and provides the TunnelsManager with functions which can send and retrieve messages or force an update of the database.

The TunnelsManager is responsible for launching and maintaining the tunnels. It also takes care of routing and network settings issues.

A user interface enables the user to modify the service policies stored in the database. Thus, they can be accessed by the

FirewallManager and the customized DNS server that filters DNS-SD requests accordingly.



Overall functional architecture.

Figure 1

4. Network architecture

4.1. Managing the tunnels

The tunnelling technology chosen for this experiment is OpenVPN with the cryptography library OpenSSL.

In OpenVPN, one end has to be a server listening to the connections of clients, which in this case are the Gplus routers of the connected homenets. A server might have several clients connected to the same network interface. Notice it can be configured such as the clients connected to the same server cannot send packets to each other. Though there might be better ways to proceed, for this experiment the choice of being server or client is made by comparing the Gplus IDs of the connected pages.

To set a tunnel with proper authentication of the other end, an architecture of OpenSSL certificates must be built. A Certificate Authority (CA) is built and owned by the server which must sign

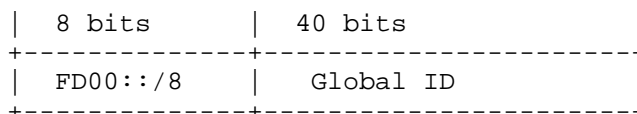
certificates to the clients. The certificates contain the Common Names of they owners, which define the identity of the tunnel endpoints. For this experiment, the Common Name of a router is its Gplus ID. Since each Gplus router may potentially host at the same time a server and multiple clients, it creates a CA and a Certificate Request (CSR). Then it publishes in Gplus a post (here called Security Settings) containing the certificate (CRT) of its CA and its CSR and makes it visible by all its circles. Therefore, when a new relationship appears in the social network, the server retrieves the client's CSR, signs it with the key of its CA and sends it back with a restricted visibility to the client. As for the client, it retrieves the CRT of the server's CA and its signed CRT. Notice there is no cryptography key sent on the social network, which is otherwise a secure channel to exchange the CAs and CSRs.

Concerning contact addresses, the Gplus router must have a globally reachable IP address whether IPv4 (for example being the CPE) or preferably IPv6. This/these addresse(s) are advertised in the Network Settings post which is published at boot time and regularly updated, and visible by all the circles of the homenet.

4.2. Configuring the network

In order to enable reachability of the devices of a connected homenet via the tunnel between them, routes must be configured. For reasons explained in Section 6, instead of injecting routes to the globally routable prefixes of the connected homenets, the described design makes the Gplus routers generate and assign ULA prefixes and only those are advertised.

In order to reduce the odds of collision, the ULA prefix is generated by the Gplus router following the following schema:



Global ID = f(hash(timestamp + GplusID))

With:

f A function that take only the 40 last bits of its argument

hash	A hashing function (SHA1 for this experiment)
timestamp	A string containing the current UNIX timestamp
GplusID	The homenet's Gplus ID
+	The string concatenation operation

Once generated, the prefix is delegated to the homenet and /64 are assigned as specified in [I-D.arkko-homenet-prefix-assignment].

On the other ends of tunnels, the ULA prefix for this homenet is retrieved from the Network Settings post in Gplus and advertised through the connected homenets by injecting AS-External-LSAs in OSPFv3.

In case there are other ULA prefixes assigned in the homenet, they should also be advertised and routed to the connected homenets. Otherwise the Default Address Selection mechanism for IPv6 specified in [RFC3484] will lead to an unpredictable behaviour as the source addressed chosen by a host to communicate over the tunnel might not be in the prefix advertised on Gplus and then would not be routed at the other end. But having other ULA prefixes is non-desirable since it increases the odds of prefix collision. In our implementation, we assume there is no other ULA prefix assigned in the homenet.

Though we strive to avoid collisions while generating the ULA prefixes, the current design assumes there is no collision and does not treat such a case. Collisions might appear in two situations: either a Gplus router chooses the same prefix as one of its connected routers, or a Gplus router has two connected routers that have the same prefix. The best solution for this is left for further study.

5. Sharing services within your set of connected homes

Connecting homenets would be pointless without any service discovery mechanism. The aim is to allow a host to query services in connected homenets, and to let only the authorized services appear in the responses.

Inside a single home, automatic service discovery is enabled by the hybrid DNS-SD proxy mechanism specified in [I-D.stenberg-homenet-dnssdext-hybrid-proxy-ospf]. The following design assumes this running on all routers of the homenet and mostly relies on it to enable service discovery over multiple homes.

Connected homenets must have distinct domain names. Each homenet must either have a domain name that is owned by their administrator or generate a local one. In case of automatic generation we again have a problem of collisions and use Gplus IDs to make them the most unlikely possible. In order to make it a minimum human-friendly too, the formatted display name of the associated Gplus page is put at the beginning, concatenated with an hyphen, 10 hexadecimal digits corresponding to the Global ID of the ULA prefix (Section 4.2) and the TLD. A generic TLD for homes might be defined in the future, though for this experiment we use ".test.".

To advertise this domain name across the homenet, the Gplus router advertises a Domain Name TLV.

To make hosts browse other homenets zones, a DNS Delegated Zone TLV must be advertise for each one of them. The S bit must be set to 0 because those zones are not full DNS-SD domains, and the B bit set to 1 so that they are recommended for browsing at b._dns-sd._udp.(domain). For each one, the domain name and authoritative DNS server address (a ULA address of the Gplus router) are retrieved from the Network Settings post published in Gplus.

Thus, the Gplus router's DNS server receives from other homes all DNS-SD queries for its home's domain name. Responses are filtered based on the source ULA address and the services authorized to the corresponding home. Notice also that A records and AAAA records that do not point to ULA addresses are dropped. A service is authorized if and only if a policy of one of the circles in which this home is allows it. For this experiment, a policy is defined as an authorized DNS-SD type of service (e.g. _http._tcp) associated to a circle, but finer granularity might be implemented (which adds complexity because of hosts changing DNS zones or name).

6. Security Considerations

The goal of the experiment is to allow homes to reach one another more easily than reaching the whole of the internet. Doing so, the boundaries of the homenet are redrawn to include multiple homes, which brings up security issues. DNS requests and most common services' connections are not encrypted, which motivates the enforcement of a secure channel between homes. Besides, tunnels also provide identity of the incoming packets.

Injecting global prefixes in other homes might be a way to advertise larger prefixes than those actually owned (e.g. advertising a /48 while only having a /56). Of course we could limit the size of advertised prefixes but this is not enough. One could imagine a PKI

verification system but this would assume support from ISPs which is not currently offered. Using ULA prefixes mitigates this issue though it adds some others (already described in Section 4.2).

Still, defining firewall rules is probably the toughest security concern. First, to prevent spoofing, only packets with source and destination addresses in the expected ULA prefixes are allowed. Even though the firewall of OpenVPN servers is not able to know for sure which connected client has sent a packet as an IP address might be spoofed, potential harm is very limited because it will not receive any packet back.

Second, relying on inability to discover unauthorized services via DNS-SD is not sufficient, hence the need to accept only traffic corresponding to authorized services. This is a non-trivial general issue since a service cannot be reduced to a contact port and IP address tuple. This issue is left for further study.

7. Experimental results

TBD

8. IANA Considerations

This document contains no request to IANA.

9. Acknowledgements

The author would like to thank Mark Townsley, Alain Fiocco, Ole Troan and Markus Stenberg for valuable mentoring of the project, as well as Pierre-Alain Dupont, Nicolas Iooss, Maico Le Pape and Guillaume Mulocher for high contribution in the design and implementation of the prototype.

10. References

10.1. Normative References

[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.

[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.cheshire-mdnsexthybrid]

Cheshire, S., "Hybrid Unicast/Multicast DNS-Based Service
Discovery", draft-cheshire-mdnsexthybrid-02 (work in
progress), July 2013.

[I-D.ietf-homenet-arch]

Chown, T., Arkko, J., Brandt, A., Troan, O., and J. Weil,
"Home Networking Architecture for IPv6",
draft-ietf-homenet-arch-09 (work in progress), July 2013.

[I-D.stenberg-homenet-dnssdext-hybrid-proxy-ospf]

Stenberg, M., "Hybrid Unicast/Multicast DNS-Based Service
Discovery Auto-Configuration Using OSPFv3",
draft-stenberg-homenet-dnssdext-hybrid-proxy-ospf-00 (work
in progress), June 2013.

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

[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.

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

[RFC6749] Hardt, D., "The OAuth 2.0 Authorization Framework",
RFC 6749, October 2012.

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

10.2. Informative References

[RFC3484] Draves, R., "Default Address Selection for Internet
Protocol version 6 (IPv6)", RFC 3484, February 2003.

Author's Address

Cedric Dessez
Cisco Systems
Paris,
France

Email: cedric@dessez.fr

Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 16, 2014

T. Chown, Ed.
University of Southampton
J. Arkko
Ericsson
A. Brandt
Sigma Designs
O. Troan
Cisco Systems, Inc.
J. Weil
Time Warner Cable
July 15, 2013

Home Networking Architecture for IPv6
draft-ietf-homenet-arch-09

Abstract

This text describes evolving networking technology within increasingly large residential home networks. The goal of this document is to define a general architecture for IPv6-based home networking, describing the associated principles, considerations and requirements. The text briefly highlights specific implications of the introduction of IPv6 for home networking, discusses the elements of the architecture, and suggests how standard IPv6 mechanisms and addressing can be employed in home networking. The architecture describes the need for specific protocol extensions for certain additional functionality. It is assumed that the IPv6 home network is not actively managed, and runs as an IPv6-only or dual-stack network. There are no recommendations in this text for the IPv4 part of the network.

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 16, 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. Introduction	4
1.1. Terminology and Abbreviations	5
2. Effects of IPv6 on Home Networking	6
2.1. Multiple subnets and routers	7
2.2. Global addressability and elimination of NAT	8
2.3. Multi-Addressing of devices	8
2.4. Unique Local Addresses (ULAs)	9
2.5. Avoiding manual configuration of IP addresses	10
2.6. IPv6-only operation	11
3. Homenet Architecture	11
3.1. General Principles	12
3.1.1. Reuse existing protocols	12
3.1.2. Minimise changes to hosts and routers	13
3.2. Homenet Topology	13
3.2.1. Supporting arbitrary topologies	13
3.2.2. Network topology models	13
3.2.3. Dual-stack topologies	18
3.2.4. Multihoming	19
3.3. A Self-Organising Network	20
3.3.1. Differentiating neighbouring homenets	21
3.3.2. Largest practical subnets	21
3.3.3. Handling varying link technologies	21
3.3.4. Homenet realms and borders	22
3.4. Homenet Addressing	23
3.4.1. Use of ISP-delegated IPv6 prefixes	23
3.4.2. Stable internal IP addresses	25
3.4.3. Internal prefix delegation	26
3.4.4. Coordination of configuration information	27
3.4.5. Privacy	28
3.5. Routing functionality	28

3.5.1.	Multicast support	29
3.5.2.	Mobility support	30
3.6.	Security	30
3.6.1.	Addressability vs reachability	30
3.6.2.	Filtering at borders	31
3.6.3.	Partial Effectiveness of NAT and Firewalls	31
3.6.4.	Device capabilities	32
3.6.5.	ULAs as a hint of connection origin	32
3.7.	Naming and Service Discovery	32
3.7.1.	Discovering services	33
3.7.2.	Assigning names to devices	34
3.7.3.	Name spaces	34
3.7.4.	The homenet name service	36
3.7.5.	Independent operation	37
3.7.6.	Considerations for LLNs	37
3.7.7.	DNS resolver discovery	37
3.7.8.	Devices roaming from the homenet	38
3.8.	Other Considerations	38
3.8.1.	Quality of Service	38
3.8.2.	Operations and Management	38
3.9.	Implementing the Architecture on IPv6	39
4.	Conclusions	39
5.	References	40
5.1.	Normative References	40
5.2.	Informative References	40
Appendix A.	Acknowledgments	43
Appendix B.	Changes	43
B.1.	Version 09 (after WGLC)	43
B.2.	Version 08	44
B.3.	Version 07	44
B.4.	Version 06	45
B.5.	Version 05	45
B.6.	Version 04	46
B.7.	Version 03	46
B.8.	Version 02	47
Authors' Addresses	48

1. Introduction

This document focuses on evolving networking technology within increasingly large residential home networks and the associated challenges with their deployment and operation. There is a growing trend in home networking for the proliferation of networking technology through an increasingly broad range of devices and media. This evolution in scale and diversity sets requirements on IETF protocols. Some of these requirements relate to the introduction of IPv6, others to the introduction of specialised networks for home automation and sensors.

While at the time of writing some complex home network topologies exist, most are relatively simple single subnet networks, and ostensibly operate using just IPv4. While there may be IPv6 traffic within the network, e.g. for service discovery, the homenet is provisioned by the ISP as an IPv4 network. Such networks also typically employ solutions that we would like to avoid, such as private [RFC1918] addressing with (cascaded) network address translation (NAT)[RFC3022], or they may require expert assistance to set up.

In contrast, emerging IPv6-capable home networks are very likely to have multiple internal subnets, e.g. to facilitate private and guest networks, heterogeneous link layers, and smart grid components, and have enough address space available to allow every device to have a globally unique address. This implies that internal routing functionality is required, and that the homenet's ISP both provides a large enough prefix to allocate a prefix to each subnet, and that a method is supported for such prefixes to be delegated efficiently to those subnets.

It is not practical to expect home users to configure their networks. Thus the assumption of this document is that the homenet is as far as possible self-organising and self-configuring, i.e. it should function without pro-active management by the residential user.

The architectural constructs in this document are focused on the problems to be solved when introducing IPv6, with an eye towards a better result than what we have today with IPv4, as well as a better result than if the IETF had not given this specific guidance. The document aims to provide the basis and guiding principles for how standard IPv6 mechanisms and addressing [RFC2460] [RFC4291] can be employed in home networking, while coexisting with existing IPv4 mechanisms. In emerging dual-stack home networks it is vital that introducing IPv6 does not adversely affect IPv4 operation. We assume that the IPv4 network architecture in home networks is what it is, and can not be modified by new recommendations. This document does

not discuss how IPv4 home networks provision or deliver support for multiple subnets. It should not be assumed that any future new functionality created with IPv6 in mind will be backward-compatible to include IPv4 support. Further, future deployments, or specific subnets within an otherwise dual-stack home network, may be IPv6-only, in which case considerations for IPv4 impact would not apply.

This document proposes a baseline homenet architecture, using protocols and implementations that are as far as possible proven and robust. The scope of the document is primarily the network layer technologies that provide the basic functionality to enable addressing, connectivity, routing, naming and service discovery. While it may, for example, state that homenet components must be simple to deploy and use, it does not discuss specific user interfaces, nor does it discuss specific physical, wireless or data-link layer considerations.

[RFC6204] defines basic requirements for customer edge routers (CERs). This document has recently been updated with the definition of requirements for specific transition tools on the CER in [I-D.ietf-v6ops-6204bis], specifically DS-Lite [RFC6333] and 6rd [RFC5969]. Such detailed specification of CER devices is considered out of scope of this architecture document, and we assume that any required update of the CER device specification as a result of adopting this architecture will be handled as separate and specific updates to these existing documents. Further, the scope of this text is the internal homenet, and thus specific features on the WAN side of the CER are out of scope for this text.

1.1. Terminology and Abbreviations

In this section we define terminology and abbreviations used throughout the text.

- o ALQDN: Ambiguous Locally Qualified Domain Name. An example would be .sitelocal.
- o Border: a point, typically resident on a router, between two networks, e.g. between the main internal homenet and a guest network. This defines point(s) at which filtering and forwarding policies for different types of traffic may be applied.
- o CER: Customer Edge Router: A border router intended for use in a homenet, which connects the homenet to a service provider network.
- o FQDN: Fully Qualified Domain Name. A globally unique name space.

- o Homenet: A home network, comprising host and router equipment, with one or more CERS providing connectivity to service provider network(s).
- o Internet Service Provider (ISP): an entity that provides access to the Internet. In this document, a service provider specifically offers Internet access using IPv6, and may also offer IPv4 Internet access. The service provider can provide such access over a variety of different transport methods such as DSL, cable, wireless, and others.
- o LLN: Low-power and lossy network.
- o LQDN: Locally Qualified Domain Name. A name space local to the homenet.
- o NAT: Network Address Translation. Typically referring to IPv4 Network Address and Port Translation (NAPT) [RFC3022].
- o NPTv6: Network Prefix Translation for IPv6 [RFC6296].
- o PCP: Port Control Protocol [I-D.ietf-pcp-base].
- o Realm: a network delimited by a defined border. A guest network within a homenet may form one realm.
- o 'Simple Security'. Defined in [RFC4864] and expanded further in [RFC6092]; describes recommended perimeter security capabilities for IPv6 networks.
- o ULA: IPv6 Unique Local Addresses [RFC4193].
- o ULQDN: Unique Locally Qualified Domain Name. An example might be .<UniqueString>.sitelocal.
- o UPnP: Universal Plug and Play. Includes the Internet Gateway Device (IGD) function, which for IPv6 is UPnP IGD Version 2 [IGD-2].
- o VM: Virtual machine.
- o WPA2: Wi-Fi Protected Access, as defined by the Wi-Fi Alliance.

2. Effects of IPv6 on Home Networking

While IPv6 resembles IPv4 in many ways, there are some notable differences in the way it may typically be deployed. It changes

address allocation principles, making multi-addressing the norm, and, through the vastly increased address space, allows globally unique IP addresses to be used for all devices in a home network. This section presents an overview of some of the key implications of the introduction of IPv6 for home networking, that are simultaneously both promising and problematic.

2.1. Multiple subnets and routers

While simple layer 3 topologies involving as few subnets as possible are preferred in home networks, the incorporation of dedicated (routed) subnets remains necessary for a variety of reasons. For instance, an increasingly common feature in modern home routers is the ability to support both guest and private network subnets. Likewise, there may be a need to separate building control or corporate extensions from the main Internet access network, or different subnets may in general be associated with parts of the homenet that have different routing and security policies. Further, link layer networking technology is poised to become more heterogeneous, as networks begin to employ both traditional Ethernet technology and link layers designed for low-power and lossy networks (LLNs), such as those used for certain types of sensor devices. Constraining the flow of certain traffic from Ethernet links to much lower capacity links thus becomes an important topic.

The introduction of IPv6 for home networking enables the potential for every home network to be delegated enough address space from its ISP to provision globally unique prefixes for each such subnet in the home. While the number of addresses in a standard /64 IPv6 prefix is practically infinite, the number of prefixes available for assignment to the home network is not. As a result the growth inhibitor for the home network shifts from the number of addresses to the number of prefixes offered by the provider; this topic is discussed in [RFC6177] (BCP 157), which recommends that "end sites always be able to obtain a reasonable amount of address space for their actual and planned usage".

The addition of routing between subnets raises a number of issues. One is a method by which prefixes can be efficiently allocated to each subnet, without user intervention. Another is the issue of how to extend mechanisms such as zero configuration service discovery which currently only operate within a single subnet using link-local traffic. In a typical IPv4 home network, there is only one subnet, so such mechanisms would normally operate as expected. For multi-subnet IPv6 home networks there are two broad choices to enable such protocols to work across the scope of the entire homenet; extend existing protocols to work across that scope, or introduce proxies for existing link layer protocols. This topic is discussed in

Section 3.7.

2.2. Global addressability and elimination of NAT

The possibility for direct end-to-end communication on the Internet to be restored by the introduction of IPv6 is on the one hand an incredible opportunity for innovation and simpler network operation, but on the other hand it is also a concern as it potentially exposes nodes in the internal networks to receipt of unwanted traffic from the Internet.

With devices and applications able to talk directly to each other when they have globally unique addresses, there may be an expectation of improved host security to compensate for this. It should be noted that many devices may (for example) ship with default settings that make them readily vulnerable to compromise by external attackers if globally accessible, or may simply not have robustness designed-in because it was either assumed such devices would only be used on private networks or the device itself doesn't have the computing power to apply the necessary security methods. In addition, the upgrade cycle for devices (or their firmware) may be slow, and/or lack auto-update mechanisms.

It is thus important to distinguish between addressability and reachability. While IPv6 offers global addressability through use of globally unique addresses in the home, whether devices are globally reachable or not would depend on any firewall or filtering configuration, and not, as is commonly the case with IPv4, the presence or use of NAT. In this respect, IPv6 networks may or may not have filters applied at their borders to control such traffic, i.e. at the homenet CER. [RFC4864] and [RFC6092] discuss such filtering, and the merits of 'default allow' against 'default deny' policies for external traffic initiated into a homenet. This document takes no position on which mode is the default, but assumes the choice for the homenet to use either mode would be available.

2.3. Multi-Addressing of devices

In an IPv6 network, devices will often acquire multiple addresses, typically at least a link-local address and one or more globally unique addresses. Where a homenet is multihomed, a device would typically receive a globally unique address (GUA) from within the delegated prefix from each upstream ISP. Devices may also have an IPv4 address if the network is dual-stack, an IPv6 Unique Local Address (ULA) [RFC4193] (see below), and one or more IPv6 Privacy Addresses [RFC4941].

It should thus be considered the norm for devices on IPv6 home

networks to be multi-addressed, and to need to make appropriate address selection decisions for the candidate source and destination address pairs for any given connection. Default Address Selection for IPv6 [RFC6724] provides a solution for this, though it may face problems in the event of multihoming where, as described above, nodes will be configured with one address from each upstream ISP prefix. In such cases the presence of upstream BCP 38 [RFC2827] ingress filtering requires multi-addressed nodes to select the correct source address to be used for the corresponding uplink. A challenge here is that the node may not have the information it needs to make that decision based on addresses alone. We discuss this challenge in Section 3.2.4.

2.4. Unique Local Addresses (ULAs)

[RFC4193] defines Unique Local Addresses (ULAs) for IPv6 that may be used to address devices within the scope of a single site. Support for ULAs for IPv6 CERNs is described in [RFC6204]. A home network running IPv6 should deploy ULAs alongside its globally unique prefix(es) to allow stable communication between devices (on different subnets) within the homenet where that externally allocated globally unique prefix may change over time, e.g. due to renumbering within the subscriber's ISP, or where external connectivity may be temporarily unavailable. A homenet using provider-assigned global addresses is exposed to its ISP renumbering the network to a much larger degree than before whereas, for IPv4, NAT isolated the user against ISP renumbering to some extent.

While setting up a network there may be a period where it has no external connectivity, in which case ULAs would be required for inter-subnet communication. In the case where LLNs are being set up in a new home/deployment (as early as during construction of the home), LLNs will likely need to use their own /48 ULA prefix. Depending upon circumstances beyond the scope of homenet, it may be impossible to renumber the ULA used by the LLN so routing between ULA /48s may be required. Also, some devices, particularly constrained devices, may have only a ULA (in addition to a link-local), while others may have both a GUA and a ULA.

Note that unlike private IPv4 RFC 1918 space, the use of ULAs does not imply use of host-based IPv6 NAT, or NPTv6 prefix-based NAT [RFC6296], rather that in an IPv6 homenet a node should use its ULA address internally, and its additional globally unique IPv6 address as a source address for external communications. By using such globally unique addresses between hosts and devices in remote networks, the architectural cost and complexity, particularly to applications, of NAT or NPTv6 translation is avoided. As such, neither IPv6 NAT or NPTv6 is recommended for use in the homenet

architecture.

Devices in a homenet may be given only a ULA as a means to restrict reachability from outside the homenet. ULAs can be used by default for devices that, without additional configuration (e.g. via a web interface), would only offer services to the internal network. For example, a printer might only accept incoming connections on a ULA until configured to be globally reachable, at which point it acquires a global IPv6 address and may be advertised via a global name space.

Where both a ULA and a global prefix are in use, the ULA source address is used to communicate with ULA destination addresses when appropriate, i.e. when the ULA source and destination lie within the /48 ULA prefix(es) known to be used within the same homenet. In cases where multiple /48 ULA prefixes are in use within a single homenet (perhaps because multiple homenet routers each independently auto-generate a /48 ULA prefix and then share prefix/routing information), utilising a ULA source address and a ULA destination address from two disjoint internal ULA prefixes is preferable to using GUAs.

While a homenet should operate correctly with two or more /48 ULAs enabled, a mechanism for the creation and use of a single /48 ULA prefix is desirable for addressing consistency and policy enforcement. It may thus be expected that one router in the homenet be elected a 'master' to delegate ULA prefixes to subnets from a single /48 ULA prefix.

A counter-argument to using ULAs is that it is undesirable to aggressively deprecate global prefixes for temporary loss of connectivity, so for a host to lose its global address there would have to be a connection breakage longer than the lease period, and even then, deprecating prefixes when there is no connectivity may not be advisable. However, it is assumed in this architecture that homenets should support and use ULAs.

2.5. Avoiding manual configuration of IP addresses

Some IPv4 home networking devices expose IPv4 addresses to users, e.g. the IPv4 address of a home IPv4 CER that may be configured via a web interface. In potentially complex future IPv6 homenets, users should not be expected to enter IPv6 literal addresses in devices or applications, given their much greater length and the apparent randomness of such addresses to a typical home user. Thus, even for the simplest of functions, simple naming and the associated (minimal, and ideally zero configuration) discovery of services is imperative for the easy deployment and use of homenet devices and applications. As mentioned previously, this means that zeroconf naming and service

discovery protocols must be capable of operating across subnet boundaries.

2.6. IPv6-only operation

It is likely that IPv6-only networking will be deployed first in 'greenfield' homenet scenarios, or perhaps as one element of an otherwise dual-stack network. Running IPv6-only adds additional requirements, e.g. for devices to get configuration information via IPv6 transport (not relying on an IPv4 protocol such as IPv4 DHCP), and for devices to be able to initiate communications to external devices that are IPv4-only. Thus, for example, the following requirements are amongst those that should be considered in IPv6-only environments:

- o Ensuring there is a way to access content in the IPv4 Internet. This can be arranged through appropriate use of NAT64 [RFC6144] and DNS64 [RFC6145], for example, or via a node-based DS-Lite [RFC6333] approach.
- o Ensuring DNS resolver discovery mechanisms are enabled for IPv6. Both stateless DHCPv6 [RFC3736] [RFC3646] and Router Advertisement options [RFC6106] may have to be supported and turned on by default to ensure maximum compatibility with all types of hosts in the network. This requires, however, that a working DNS server is known and addressable via IPv6, and that the automatic discovery of such a server is possible through multiple routers in the homenet.
- o Ensuring all nodes in the home network support operations in IPv6-only mode. Some current devices work well with dual-stack but fail to recognise connectivity when IPv4 DHCP fails, for instance.

The widespread availability of robust solutions to these types of requirements will help accelerate the uptake of IPv6-only homenets. The specifics of these are however beyond the scope of this document, especially those functions that reside on the CER.

3. Homenet Architecture

The aim of this text is to outline how to construct advanced IPv6-based home networks involving multiple routers and subnets using standard IPv6 protocols and addressing [RFC2460] [RFC4291]. In this section, we present the elements of the proposed home networking architecture, with discussion of the associated design principles.

In general, home network equipment needs to be able to operate in

networks with a range of different properties and topologies, where home users may plug components together in arbitrary ways and expect the resulting network to operate. Significant manual configuration is rarely, if at all, possible, or even desirable given the knowledge level of typical home users. Thus the network should, as far as possible, be self-configuring, though configuration by advanced users should not be precluded.

The homenet needs to be able to handle or provision at least

- o Routing
- o Prefix configuration for routers
- o Name resolution
- o Service discovery
- o Network security

The remainder of this document describes the principles by which the homenet architecture may deliver these properties.

3.1. General Principles

There is little that the Internet standards community can do about the physical topologies or the need for some networks to be separated at the network layer for policy or link layer compatibility reasons. However, there is a lot of flexibility in using IP addressing and inter-networking mechanisms. This text discusses how such flexibility should be used to provide the best user experience and ensure that the network can evolve with new applications in the future. The principles described in this text should be followed when designing homenet solutions.

3.1.1. Reuse existing protocols

It is desirable to reuse existing protocols where possible, but at the same time to avoid consciously precluding the introduction of new or emerging protocols. A generally conservative approach, giving weight to running (and available) code, is preferable. Where new protocols are required, evidence of commitment to implementation by appropriate vendors or development communities is highly desirable. Protocols used should be backwardly compatible, and forward compatible where changes are made.

3.1.2. Minimise changes to hosts and routers

Where possible, any requirement for changes to hosts and routers should be minimised, though solutions which, for example, incrementally improve capability with host or router changes may be acceptable.

3.2. Homenet Topology

This section considers homenet topologies, and the principles that may be applied in designing an architecture to support as wide a range of such topologies as possible.

3.2.1. Supporting arbitrary topologies

There should ideally be no built-in assumptions about the topology in home networks, as users are capable of connecting their devices in 'ingenious' ways. Thus arbitrary topologies and arbitrary routing will need to be supported, or at least the failure mode for when the user makes a mistake should be as robust as possible, e.g. de-activating a certain part of the infrastructure to allow the rest to operate. In such cases, the user should ideally have some useful indication of the failure mode encountered.

There should be no topology scenarios which cause loss of connectivity, except when the user creates a physical island within the topology. Some potentially pathological cases that can be created include bridging ports of a router together, however this case can be detected and dealt with by the router. Loops within a routed topology are in a sense good in that they offer redundancy. Bridging loops can be dangerous but are also detectable when a switch learns the MAC of one of its interfaces on another or runs a spanning tree or link state protocol. It is only loops using simple repeaters that are truly pathological.

The topology of the homenet may change over time, due to the addition or removal of equipment, but also due to temporary failures or connectivity problems. In some cases this may lead to, for example, a multihomed homenet being split into two isolated homenets, or, after such a fault is remedied, two isolated parts reconfiguring back to a single network.

3.2.2. Network topology models

Most IPv4 home network models at the time of writing tend to be relatively simple, typically a single NAT router to the ISP and a single internal subnet but, as discussed earlier, evolution in network architectures is driving more complex topologies, such as the

separation of guest and private networks. There may also be some cascaded IPv4 NAT scenarios, which we mention in the next section. For IPv6 homenets, the network models described in [RFC6204] and its successor RFC 6204-bis [I-D.ietf-v6ops-6204bis] should, as a minimum, be supported.

There are a number of properties or attributes of a home network that we can use to describe its topology and operation. The following properties apply to any IPv6 home network:

- o Presence of internal routers. The homenet may have one or more internal routers, or may only provide subnetting from interfaces on the CER.
- o Presence of isolated internal subnets. There may be isolated internal subnets, with no direct connectivity between them within the homenet (with each having its own external connectivity). Isolation may be physical, or implemented via IEEE 802.1q VLANs. The latter is however not something a typical user would be expected to configure.
- o Demarcation of the CER. The CER(s) may or may not be managed by the ISP. If the demarcation point is such that the customer can provide or manage the CER, its configuration must be simple. Both models must be supported.

Various forms of multihoming are likely to become more prevalent with IPv6 home networks, where the homenet may have two or more external ISP connections, as discussed further below. Thus the following properties should also be considered for such networks:

- o Number of upstream providers. The majority of home networks today consist of a single upstream ISP, but it may become more common in the future for there to be multiple ISPs, whether for resilience or provision of additional services. Each would offer its own prefix. Some may or may not provide a default route to the public Internet.
- o Number of CERs. The homenet may have a single CER, which might be used for one or more providers, or multiple CERs. The presence of multiple CERs adds additional complexity for multihoming scenarios, and protocols like PCP that need to manage connection-oriented state mappings.

In the following sections we give some examples of the types of homenet topologies we may see in the future. This is not intended to be an exhaustive or complete list, rather an indicative one to facilitate the discussion in this text.

3.2.2.1. A: Single ISP, Single CER, Internal routers

Figure 1 shows a home network with multiple local area networks. These may be needed for reasons relating to different link layer technologies in use or for policy reasons, e.g. classic Ethernet in one subnet and a LLN link layer technology in another. In this example there is no single router that a priori understands the entire topology. The topology itself may also be complex, and it may not be possible to assume a pure tree form, for instance (because home users may plug routers together to form arbitrary topologies including loops).

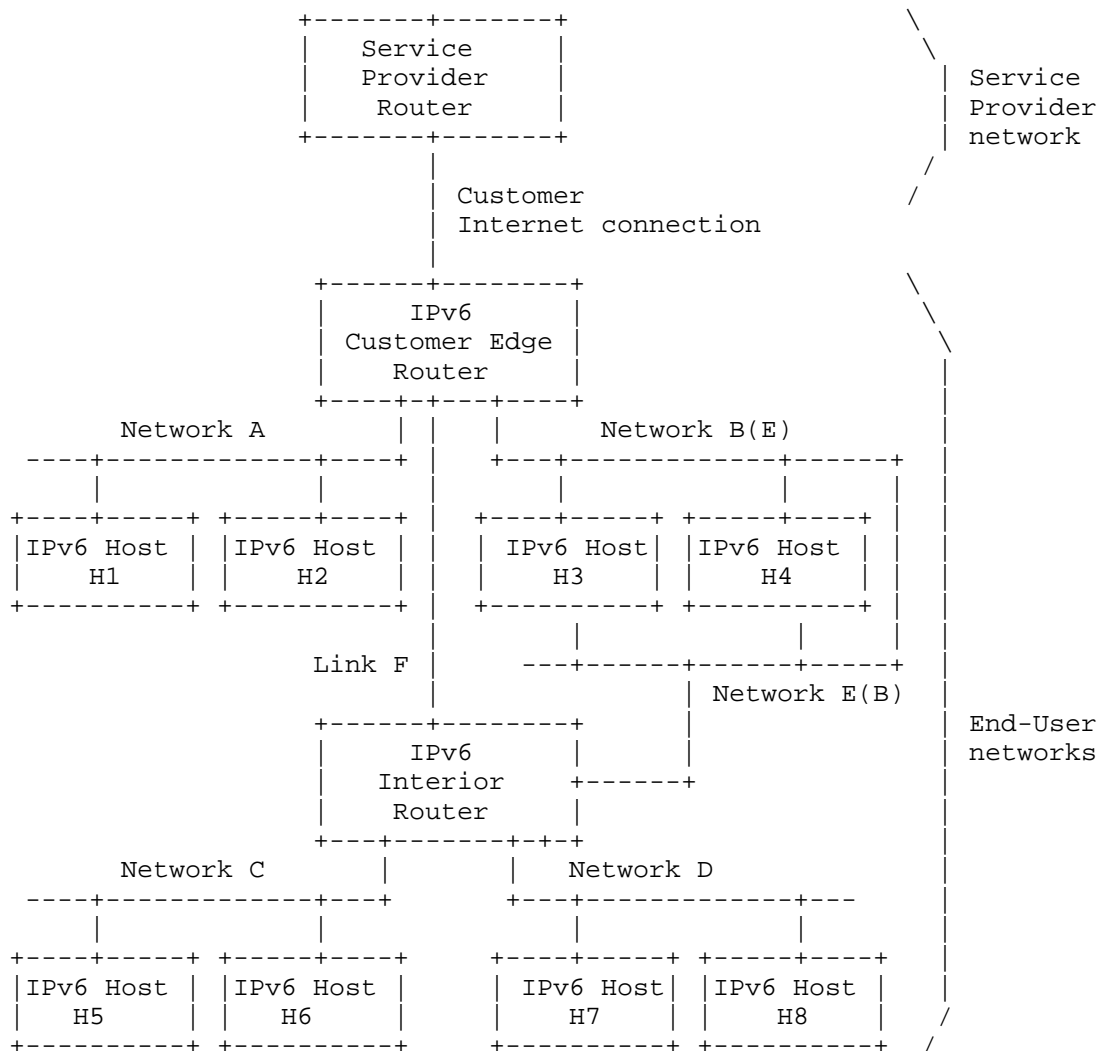


Figure 1

In this diagram there is one CER. It has a single uplink interface. It has three additional interfaces connected to Network A, Link F, and Network B. IPv6 Internal Router (IR) has four interfaces connected to Link F, Network C, Network D and Network E. Network B and Network E have been bridged, likely inadvertently. This could be as a result of connecting a wire between a switch for Network B and a switch for Network E.

Any of logical Networks A through F might be wired or wireless.

Where multiple hosts are shown, this might be through one or more physical ports on the CER or IPv6 (IR), wireless networks, or through one or more layer-2 only Ethernet switches.

3.2.2.2. B: Two ISPs, Two CERs, Shared subnet

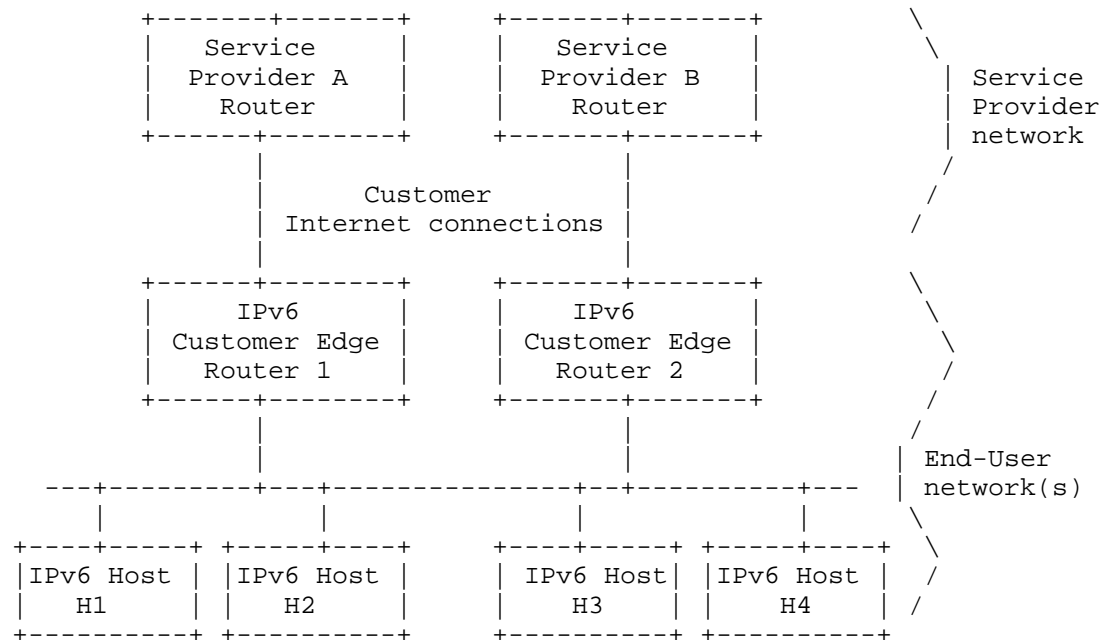


Figure 2

Figure 2 illustrates a multihomed homenet model, where the customer has connectivity via CER1 to ISP A and via CER2 to ISP B. This example shows one shared subnet where IPv6 nodes would potentially be multihomed and receive multiple IPv6 global addresses, one per ISP. This model may also be combined with that shown in Figure 1 to create a more complex scenario with multiple internal routers. Or the above shared subnet may be split in two, such that each CER serves a separate isolated subnet, which is a scenario seen with some IPv4 networks today.

3.2.2.3. C: Two ISPs, One CER, Shared subnet

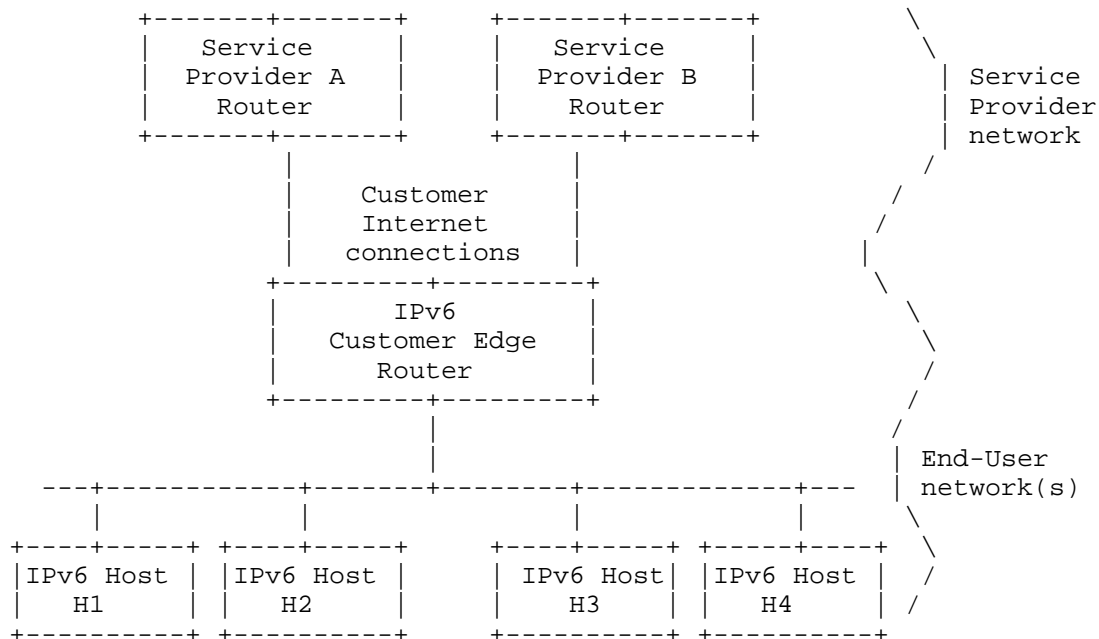


Figure 3

Figure 3 illustrates a model where a home network may have multiple connections to multiple providers or multiple logical connections to the same provider, with shared internal subnets.

In general, while the architecture may focus on likely common topologies, it should not preclude any arbitrary topology from being constructed.

3.2.3. Dual-stack topologies

It is expected that most homenet deployments will for the immediate future be dual-stack IPv4/IPv6. In such networks it is important not to introduce new IPv6 capabilities that would cause a failure if used alongside IPv4+NAT, given that such dual-stack homenets will be commonplace for some time. That said, it is desirable that IPv6 works better than IPv4 in as many scenarios as possible. Further, the homenet architecture must operate in the absence of IPv4.

A general recommendation is to follow the same topology for IPv6 as is used for IPv4, but not to use NAT. Thus there should be routed

IPv6 where an IPv4 NAT is used and, where there is no NAT, routing or bridging may be used. Routing may have advantages when compared to bridging together high speed and lower speed shared media, and in addition bridging may not be suitable for some networks, such as ad-hoc mobile networks.

In some cases IPv4 home networks may feature cascaded NATs. End users are frequently unaware that they have created such networks as 'home routers' and 'home switches' are frequently confused. In addition, there are cases where NAT routers are included within Virtual Machine Hypervisors, or where Internet connection sharing services have been enabled. This document applies equally to such hidden NAT 'routers'. IPv6 routed versions of such cases will be required. We should thus also note that routers in the homenet may not be separate physical devices; they may be embedded within other devices.

3.2.4. Multihoming

A homenet may be multihomed to multiple providers, as the network models above illustrate. This may either take a form where there are multiple isolated networks within the home or a more integrated network where the connectivity selection needs to be dynamic. Current practice is typically of the former kind, but the latter is expected to become more commonplace.

In the general homenet architecture, multihomed hosts should be multi-addressed with a global IPv6 address from the global prefix delegated from each ISP they communicate with or through. When such multi-addressing is in use, hosts need some way to pick source and destination address pairs for connections. A host may choose a source address to use by various methods, most commonly [RFC6724]. Applications may of course do different things, and this should not be precluded.

For the single CER Network Model C illustrated above, multihoming may be offered by source routing at the CER. With multiple exit routers, as in CER Network Model B, the complexity rises. Given a packet with a source address on the home network, the packet must be routed to the proper egress to avoid BCP 38 filtering if exiting through the wrong ISP. It is highly desirable that the packet is routed in the most efficient manner to the correct exit, though as a minimum requirement the packet should not be dropped.

The homenet architecture should support both the above models, i.e. one or more CERs. However, the general multihoming problem is broad, and solutions suggested to date within the IETF have included complex architectures for monitoring connectivity, traffic engineering,

identifier-locator separation, connection survivability across multihoming events, and so on. It is thus important that the homenet architecture should as far as possible minimise the complexity of any multihoming support.

An example of such a 'simpler' approach has been documented in [I-D.ietf-v6ops-ipv6-multihoming-without-ipv6nat]. Alternatively a flooding/routing protocol could potentially be used to pass information through the homenet, such that internal routers and ultimately end hosts could learn per-prefix configuration information, allowing better address selection decisions to be made. However, this would imply router and, most likely, host changes. Another avenue is to introduce support for source routing throughout the homenet; while greatly improving the 'intelligence' of routing decisions within the homenet, such an approach would require relatively significant router changes but avoid host changes.

As explained previously, while NPTv6 has been proposed for providing multi-homing support in networks, its use is not recommended in the homenet architecture.

It should be noted that some multihoming scenarios may see one upstream being a "walled garden", and thus only appropriate for connectivity to the services of that provider; an example may be a VPN service that only routes back to the enterprise business network of a user in the homenet. While we should not specifically target walled garden multihoming as a principal goal, it should not be precluded.

The homenet architecture should also not preclude use of host or application-oriented tools, e.g. Shim6 [RFC5533], MPTCP [RFC6824] or Happy Eyeballs [RFC6555]. In general, any incremental improvements obtained by host changes should give benefit for the hosts introducing them, but not be required.

3.3. A Self-Organising Network

The home network architecture should be naturally self-organising and self-configuring under different circumstances relating to the connectivity status to the Internet, number of devices, and physical topology. At the same time, it should be possible for advanced users to manually adjust (override) the current configuration.

While a goal of the homenet architecture is for the network to be as self-organising as possible, there may be instances where some manual configuration is required, e.g. the entry of a cryptographic key to apply wireless security, or to configure a shared routing secret. The latter may be relevant when considering how to bootstrap a

routing configuration. It is highly desirable that the number of such configurations is minimised.

3.3.1. Differentiating neighbouring homenets

It is important that self-configuration with 'unintended' devices is avoided. There should be a way for a user to administratively assert in a simple way whether or not a device belongs to a homenet. The goal is to allow the establishment of borders, particularly between two adjacent homenets, and to avoid unauthorised devices from participating in the homenet. Such an authorisation capability may need to operate through multiple hops in the homenet.

The homenet should thus support a way for a homenet owner to claim ownership of their devices in a reasonably secure way. This could be achieved by a pairing mechanism, by for example pressing buttons simultaneously on an authenticated and a new homenet device, or by an enrolment process as part of an autonomic networking environment.

3.3.2. Largest practical subnets

Today's IPv4 home networks generally have a single subnet, and early dual-stack deployments have a single congruent IPv6 subnet, possibly with some bridging functionality. More recently, some vendors have started to introduce 'home' and 'guest' functions, which in IPv6 would be implemented as two subnets.

Future home networks are highly likely to have one or more internal routers and thus need multiple subnets, for the reasons described earlier. As part of the self-organisation of the network, the homenet should subdivide itself to the largest practical subnets that can be constructed within the constraints of link layer mechanisms, bridging, physical connectivity, and policy, and where applicable performance or other criteria. In such subdivisions the logical topology may not necessarily match the physical topology. This text does not, however, make recommendations on how such subdivision should occur. It is expected that subsequent documents will address this problem.

While it may be desirable to maximise the chance of link-local protocols operating across a homenet by maximising the size of a subnet, multi-subnet home networks are inevitable, so their support must be included.

3.3.3. Handling varying link technologies

Homenets tend to grow organically over many years, and a homenet will typically be built over link-layer technologies from different

generations. Current homenets typically use links ranging from 1Mbit/s up to 1Gbit/s, which is a three orders of magnitude throughput discrepancy. We expect this discrepancy to widen further as both high-speed and low-power technologies are deployed.

Homenet protocols should be designed to deal well with interconnecting links of very different throughputs. In particular, flows local to a link should not be flooded throughout the homenet, even when sent over multicast, and, whenever possible, the homenet protocols should be able to choose the faster links and avoid the slower ones.

Links (particularly wireless links) may also have limited numbers of transmit opportunities (txops), and there is a clear trend driven by both power and downward compatibility constraints toward aggregation of packets into these limited txops while increasing throughput. Transmit opportunities may be a system's scarcest resource and therefore also strongly limit actual throughput available.

Therefore protocols that avoid being 'chatty', do not require flooding, and enable isolation of traffic between subnets are preferable to those which burn scarce resources.

3.3.4. Homenet realms and borders

The homenet will need to be aware of the extent of its own 'site', which will, for example, define the borders for ULA and site scope multicast traffic, and may require specific security policies to be applied. The homenet will have one or more such borders with external connectivity providers.

A homenet will most likely also have internal borders between internal realms, e.g. a guest realm or a corporate network extension realm. It should be possible to automatically discover these borders, which will determine, for example, the scope of where network prefixes, routing information, network traffic, service discovery and naming may be shared. The default mode internally should be to share everything.

It is expected that a realm would span at least an entire subnet, and thus the borders lie at routers which receive delegated prefixes within the homenet. It is also desirable for a richer security model that hosts, which may be running in a transparent communication mode, are able to make communication decisions based on available realm and associated prefix information in the same way that routers at realm borders can.

A simple homenet model may just consider three types of realm and the

borders between them, namely the internal homenet, the ISP and a guest network. In this case the borders will include that from the homenet to the ISP, that from the guest network to the ISP, and that from the homenet to the guest network. Regardless, it should be possible for additional types of realms and borders to be defined, e.g. for some specific Grid or LLN-based network, and for these to be detected automatically, and for an appropriate default policy to be applied as to what type of traffic/data can flow across such borders.

It is desirable to classify the external border of the home network as a unique logical interface separating the home network from service provider network/s. This border interface may be a single physical interface to a single service provider, multiple layer 2 sub-interfaces to a single service provider, or multiple connections to a single or multiple providers. This border makes it possible to describe edge operations and interface requirements across multiple functional areas including security, routing, service discovery, and router discovery.

It should be possible for the homenet user to override any automatically determined borders and the default policies applied between them.

3.4. Homenet Addressing

The IPv6 addressing scheme used within a homenet must conform to the IPv6 addressing architecture [RFC4291]. In this section we discuss how the homenet needs to adapt to the prefixes made available to it by its upstream ISP, such that internal subnets, hosts and devices can obtain the and configure the necessary addressing information to operate.

3.4.1. Use of ISP-delegated IPv6 prefixes

Discussion of IPv6 prefix allocation policies is included in [RFC6177]. In practice, a homenet may receive an arbitrary length IPv6 prefix from its provider, e.g. /60, /56 or /48. The offered prefix may be stable or change from time to time; it is generally expected that ISPs will offer relatively stable prefixes to their residential customers. Regardless, the home network needs to be adaptable as far as possible to ISP prefix allocation policies, and thus make no assumptions about the stability of the prefix received from an ISP, or the length of the prefix that may be offered.

However, if, for example, only a /64 is offered by the ISP, the homenet may be severely constrained or even unable to function. [RFC6177] (BCP 157) states that "a key principle for address management is that end sites always be able to obtain a reasonable

amount of address space for their actual and planned usage, and over time ranges specified in years rather than just months. In practice, that means at least one /64, and in most cases significantly more. One particular situation that must be avoided is having an end site feel compelled to use IPv6-to-IPv6 Network Address Translation or other burdensome address conservation techniques because it could not get sufficient address space." This architecture text assumes that this guidance is being followed by ISPs.

There are many problems that would arise from a homenet not being offered a sufficient prefix size for its needs. Rather than attempt to contrive a method for a homenet to operate in a constrained manner when faced with insufficient prefixes, such as the use of subnet prefixes longer than /64 (which would break SLAAC), use of NPTv6, or falling back to bridging across potentially very different media, it is recommended that the receiving router instead enters an error state and issues appropriate warnings. Some consideration may need to be given to how such a warning or error state should best be presented to a typical home user.

Thus a homenet CER should request, for example via DHCP-PD, that it would like a /48 prefix from its ISP, i.e. it asks the ISP for the maximum size prefix it might expect to be offered, even if in practice it may only be offered a /56 or /60. For a typical IPv6 homenet, it is not recommended that an ISP offer less than a /60 prefix, and it is highly preferable that the ISP offers at least a /56. It is expected that the allocated prefix to the homenet from any single ISP is a contiguous, aggregated one. While it may be possible for a homenet CER to issue multiple prefix requests to attempt to obtain multiple delegations, such behaviour is out of scope of this document.

The norm for residential customers of large ISPs may be similar to their single IPv4 address provision; by default it is likely to remain persistent for some time, but changes in the ISP's own provisioning systems may lead to the customer's IP (and in the IPv6 case their prefix pool) changing. It is not expected that ISPs will generally support Provider Independent (PI) addressing for residential homenets.

When an ISP does need to restructure, and in doing so renumber its customer homenets, 'flash' renumbering is likely to be imposed. This implies a need for the homenet to be able to handle a sudden renumbering event which, unlike the process described in [RFC4192], would be a 'flag day' event, which means that a graceful renumbering process moving through a state with two active prefixes in use would not be possible. While renumbering can be viewed as an extended version of an initial numbering process, the difference between flash

renumbering and an initial 'cold start' is the need to provide service continuity.

There may be cases where local law means some ISPs are required to change IPv6 prefixes (current IPv4 addresses) for privacy reasons for their customers. In such cases it may be possible to avoid an instant 'flash' renumbering and plan a non-flag day renumbering as per RFC 4192. Similarly, if an ISP has a planned renumbering process, it may be able to adjust lease timers, etc appropriately.

The customer may of course also choose to move to a new ISP, and thus begin using a new prefix. In such cases the customer should expect a discontinuity, and not only may the prefix change, but potentially also the prefix length if the new ISP offers a different default size prefix. The homenet may also be forced to renumber itself if significant internal 'replumbing' is undertaken by the user. Regardless, it's desirable that homenet protocols support rapid renumbering and that operational processes don't add unnecessary complexity for the renumbering process. Further, the introduction of any new homenet protocols should not make any form of renumbering any more complex than it already is.

Finally, the internal operation of the home network should also not depend on the availability of the ISP network at any given time, other than of course for connectivity to services or systems off the home network. This reinforces the use of ULAs for stable internal communication, and the need for a naming and service discovery mechanism that can operate independently within the homenet.

3.4.2. Stable internal IP addresses

The network should by default attempt to provide IP-layer connectivity between all internal parts of the homenet as well as to and from the external Internet, subject to the filtering policies or other policy constraints discussed later in the security section.

ULAs should be used within the scope of a homenet to support stable routing and connectivity between subnets and hosts regardless of whether a globally unique ISP-provided prefix is available. In the case of a prolonged external connectivity outage, ULAs allow internal operations across routed subnets to continue. ULA addresses also allow constrained LLN devices to create permanent relationships between IPv6 addresses, e.g. from a wall controller to a lamp, where symbolic host names would require additional non-volatile memory and updating global prefixes in sleeping LLN devices might also be problematic.

As discussed previously, it would be expected that ULAs would

normally be used alongside one or more global prefixes in a homenet, such that hosts become multi-addressed with both globally unique and ULA prefixes. ULAs should be used for all devices, not just those intended to only have internal connectivity. Default address selection would then enable ULAs to be preferred for internal communications between devices that are using ULA prefixes generated within the same homenet.

In cases where ULA prefixes are in use within a homenet but there is no external IPv6 connectivity (and thus no GUAs in use), recommendations ULA-5, L-3 and L-4 in RFC 6204 should be followed to ensure correct operation, in particular where the homenet may be dual-stack with IPv4 external connectivity. The use of the Route Information Option described in [RFC4191] provides a mechanism to advertise such more-specific ULA routes.

The use of ULAs should be restricted to the homenet scope through filtering at the border(s) of the homenet, as mandated by RFC 6024 requirement S-2.

Note that it is possible that in some cases multiple /48 ULA prefixes may be in use within the same homenet, e.g. when the network is being deployed, perhaps also without external connectivity. In cases where multiple ULA /48's are in use, hosts need to know that each /48 is local to the homenet, e.g. by inclusion in their local address selection policy table.

3.4.3. Internal prefix delegation

As mentioned above, there are various sources of prefixes. From the homenet perspective, a single global prefix from each ISP should be received on the border CER [RFC3633]. Where multiple CERs exist with multiple ISP prefix pools, it is expected that routers within the homenet would assign themselves prefixes from each ISP they communicate with/through. As discussed above, a ULA prefix should be provisioned for stable internal communications or for use on constrained/LLN networks.

The delegation or availability of a prefix pool to the homenet should allow subsequent internal autonomous delegation of prefixes for use within the homenet. Such internal delegation should not assume a flat or hierarchical model, nor should it make an assumption about whether the delegation of internal prefixes is distributed or centralised. The assignment mechanism should provide reasonable efficiency, so that typical home network prefix allocation sizes can accommodate all the necessary /64 allocations in most cases, and not waste prefixes. Further, duplicate assignment of multiple /64s to the same network should be avoided, and the network should behave as

gracefully as possible in the event of prefix exhaustion (though the options in such cases may be limited).

Where the home network has multiple CERs and these are delegated prefix pools from their attached ISPs, the internal prefix delegation would be expected to be served by each CER for each prefix associated with it. However, where ULAs are used, most likely in parallel with global prefixes, one router should be elected as 'master' for delegation of ULA prefixes for the homenet, such that only one /48 ULA covers the whole homenet where possible. That router should generate a /48 ULA for the site, and then delegate /64's from that ULA prefix to subnets. In cases where two /48 ULAs are generated within a homenet, the network should still continue to function, meaning that hosts will need to determine that each ULA is local to the homenet.

Delegation within the homenet should result in each link being assigned a stable prefix that is persistent across reboots, power outages and similar short-term outages. The availability of persistent prefixes should not depend on the router boot order. The addition of a new routing device should not affect existing persistent prefixes, but persistence may not be expected in the face of significant 'replumbing' of the homenet. However, delegated ULA prefixes within the homenet should remain persistent through an ISP-driven renumbering event.

Provisioning such persistent prefixes may imply the need for stable storage on routing devices, and also a method for a home user to 'reset' the stored prefix should a significant reconfiguration be required (though ideally the home user should not be involved at all).

There are multiple potential solutions for prefix delegation within a homenet. One solution could be based on DHCPv6 PD, as described in [RFC3315] and [RFC3633]. An alternative solution could be to convey prefixes within the chosen homenet routing protocol. This document makes no specific recommendation, but notes that it is very likely that all routing devices participating in a homenet must use the same internal prefix delegation method. This implies that only one delegation method should be in use.

3.4.4. Coordination of configuration information

The network elements will need to be integrated in a way that takes account of the various lifetimes on timers that are used on different elements, e.g. DHCPv6 PD, router, valid prefix and preferred prefix timers.

3.4.5. Privacy

There are no specific privacy concerns discussed in this text. If ISPs offer relatively stable IPv6 prefixes to customers, the network prefix part of addresses associated with the homenet may not change over a reasonably long period of time. This exposure is similar to IPv4 networks using NAT, where the IPv4 address received from the ISP may change over time, but not necessarily that frequently.

Hosts inside an IPv6 homenet may get new IPv6 addresses over time regardless, e.g. through Privacy Addresses [RFC4941]. This may benefit mutual privacy of users within a home network, but not mask which home network traffic is sourced from.

3.5. Routing functionality

Routing functionality is required when there are multiple routers deployed within the internal home network. This functionality could be as simple as the current 'default route is up' model of IPv4 NAT, or, more likely, it would involve running an appropriate routing protocol. Regardless of the solution method, the functionality discussed below should be met.

The homenet unicast routing protocol should be a previously deployed protocol that has been shown to be reliable, robust, to allow lightweight implementations, and of which open source implementations are available. It is desirable, but not absolutely required, that the routing protocol be able to give a complete view of the network, and that it be able to pass around more than just routing information.

Multiple interface PHYs must be accounted for in the homenet routed topology. Technologies such as Ethernet, WiFi, MoCA, etc must be capable of coexisting in the same environment and should be treated as part of any routed deployment. The inclusion of the PHY layer characteristics including bandwidth, loss, and latency in path computation should be considered for optimising communication in the homenet.

The routing protocol should support the generic use of multiple customer Internet connections, and the concurrent use of multiple delegated prefixes. A routing protocol that can make routing decisions based on source and destination addresses is thus desirable, to avoid upstream ISP BCP38 ingress filtering problems. Multihoming support should also include load-balancing to multiple providers, and failover from a primary to a backup link when available. The protocol however should not require upstream ISP connectivity to be established to continue routing within the

homenet.

The routing environment should be self-configuring, as discussed previously. An example of how OSPFv3 can be self-configuring in a homenet is described in [I-D.ietf-ospf-ospfv3-autoconfig]. Minimising convergence time should be a goal in any routed environment, but as a guideline a maximum convergence time at most 30 seconds should be the target.

As per prefix delegation, it is assumed that a single routing solution is in use in the homenet architecture. If there is an identified need to support multiple solutions, these must be interoperable.

An appropriate mechanism is required to discover which router(s) in the homenet are providing the CER function. Borders may include but are not limited to the interface to the upstream ISP, a gateway device to a separate home network such as a LLN network, or a gateway to a guest or private corporate extension network. In some cases there may be no border present, which may for example occur before an upstream connection has been established. The border discovery functionality may be integrated into the routing protocol itself, but may also be imported via a separate discovery mechanism.

In general, LLN or other networks should be able to attach and participate the same way as the main homenet, or alternatively map/be gatewayed to the main homenet. Current home deployments use largely different mechanisms in sensor and basic Internet connectivity networks. IPv6 VM solutions may also add additional routing requirements.

3.5.1. Multicast support

It is desirable that, subject to the capacities of devices on certain media types, multicast routing is supported across the homenet. The natural scopes for internal multicast would be link-local or site-local, with the latter constrained within the homenet, but other policy borders, e.g. to a guest subnet, or to certain media types, may also affect where specific multicast traffic is routed.

There may be different drivers for multicast to be supported across the homenet, e.g. for homenet-wide service discovery should a site-scope multicast service discovery protocol be defined, or potentially for novel streaming or filesharing applications. Where multicast is routed across a homenet an appropriate multicast routing protocol is required, one that as per the unicast routing protocol should be self-configuring. It must be possible to scope or filter multicast traffic to avoid it being flooded to network media where devices

cannot reasonably support it.

Multicast may also be received by or sourced from the homenet from/to external networks, e.g. where video applications use multicast to conserve the bandwidth they consume. Such multicast traffic would be greater than site scope.

The multicast environment should support the ability for applications to pick a unique multicast group to use.

3.5.2. Mobility support

Devices may be mobile within the homenet. While resident on the same subnet, their address will remain persistent, but should devices move to a different (wireless) subnet, they will acquire a new address in that subnet. It is desirable that the homenet supports internal device mobility. To do so, the homenet may either extend the reach of specific wireless subnets to enable wireless roaming across the home (availability of a specific subnet across the home), or it may support mobility protocols to facilitate such roaming where multiple subnets are used.

3.6. Security

The security of an IPv6 homenet is an important consideration. The most notable difference to the IPv4 operational model is the removal of NAT, the introduction of global addressability of devices, and thus a need to consider whether devices should have global reachability. Regardless, hosts need to be able to operate securely, end-to-end where required, and also be robust against malicious traffic directed towards them. However, there are other challenges introduced, e.g. default filtering policies at the borders between various homenet realms.

3.6.1. Addressability vs reachability

An IPv6-based home network architecture should embrace the transparent end-to-end communications model as described in [RFC2775]. Each device should be globally addressable, and those addresses must not be altered in transit. However, security perimeters can be applied to restrict end-to-end communications, and thus while a host may be globally addressable it may not be globally reachable.

[RFC4864] describes a 'Simple Security' model for IPv6 networks, whereby stateful perimeter filtering can be applied to control the reachability of devices in a homenet. RFC 4864 states in Section 4.2 that "the use of firewalls ... is recommended for those that want

boundary protection in addition to host defences". It should be noted that a 'default deny' filtering approach would effectively replace the need for IPv4 NAT traversal protocols with a need to use a signalling protocol to request a firewall hole be opened, e.g. a protocol such as UPnP or PCP [I-D.ietf-pcp-base]. In networks with multiple CERS, the signalling would need to handle the cases of flows that may use one or more exit routers. CERS would need to be able to advertise their existence for such protocols.

[RFC6092] expands on RFC 4864, giving a more detailed discussion of IPv6 perimeter security recommendations, without mandating a 'default deny' approach. Indeed, RFC 6092 does not enforce a particular mode of operation, instead stating that CERS must provide an easily selected configuration option that permits a 'transparent' mode, thus ensuring a 'default allow' model is available. The homenet architecture text makes no recommendation on the default setting, and refers the reader to RFC 6092.

3.6.2. Filtering at borders

It is desirable that there are mechanisms to detect different types of borders within the homenet, as discussed previously, and further mechanisms to then apply different types of filtering policies at those borders, e.g. whether naming and service discovery should pass a given border. Any such policies should be able to be easily applied by typical home users, e.g. to give a user in a guest network access to media services in the home, or access to a printer. Simple mechanisms to apply policy changes, or associations between devices, will be required.

There are cases where full internal connectivity may not be desirable, e.g. in certain utility networking scenarios, or where filtering is required for policy reasons against guest network subnet(s). Some scenarios/models may as a result involve running isolated subnet(s) with their own CERS. In such cases connectivity would only be expected within each isolated network (though traffic may potentially pass between them via external providers).

LLNs provide an another example of where there may be secure perimeters inside the homenet. Constrained LLN nodes may implement network key security but may depend on access policies enforced by the LLN border router.

3.6.3. Partial Effectiveness of NAT and Firewalls

Security by way of obscurity (address translation) or through firewalls (filtering) is at best only partially effective. The very poor security track record of home computer, home networking and

business PC computers and networking is testimony to this. A security compromise behind the firewall of any device exposes all others, making an entire network that relies on obscurity or a firewall as vulnerable as the most insecure device on the private side of the network.

However, given current evidence of home network products with very poor default device security, putting a firewall in place does provide some level of protection. The use of firewalls today, whether a good practice or not, is common practice and whatever protection afforded, even if marginally effective, should not be lost. Thus, while it is highly desirable that all hosts in a homenet be adequately protected by built-in security functions, it should also be assumed that all CERs will continue to support appropriate perimeter defence functions, as per [I-D.ietf-v6ops-6204bis].

3.6.4. Device capabilities

In terms of the devices, homenet hosts should implement their own security policies in accordance to their computing capabilities. They should have the means to request transparent communications to be able to be initiated to them through security filters in the homenet, either for all ports or for specific services. Users should have simple methods to associate devices to services that they wish to operate transparently through (CER) borders.

3.6.5. ULAs as a hint of connection origin

As noted in Section 3.6, if appropriate filtering is in place on the CER(s), as mandated by RFC 6024 requirement S-2, a ULA source address may be taken as an indication of locally sourced traffic. This indication could then be used with security settings to designate between which nodes a particular application is allowed to communicate, provided ULA address space is filtered appropriately at the boundary of the realm.

3.7. Naming and Service Discovery

The homenet requires devices to be able to determine and use unique names by which they can be accessed on the network. Users and devices will need to be able to discover devices and services available on the network, e.g. media servers, printers, displays or specific home automation devices. Thus naming and service discovery must be supported in the homenet, and, given the nature of typical home network users, the service(s) providing this function must as far as possible support unmanaged operation.

The naming system will be required to work internally or externally,

be the user within the homenet or outside it, i.e. the user should be able to refer to devices by name, and potentially connect to them, wherever they may be. The most natural way to think about such naming and service discovery is to enable it to work across the entire homenet residence (site), disregarding technical borders such as subnets but respecting policy borders such as those between guest and other internal network realms. Remote access may be desired by the homenet residents while travelling, but also potentially by manufacturers or other 'benevolent' third parties.

3.7.1. Discovering services

Users will typically perform service discovery through GUI interfaces that allow them to browse services on their network in an appropriate and intuitive way. Devices may also need to discover other devices, without any user intervention or choice. Either way, such interfaces are beyond the scope of this document, but the interface should have an appropriate API for the discovery to be performed.

Such interfaces may also typically hide the local domain name element from users, especially where only one name space is available. However, as we discuss below, in some cases the ability to discover available domains may be useful.

We note that current zero-configuration service discovery protocols are generally aimed at single subnets. There is thus a choice to make for multi-subnet homenets as to whether such protocols should be proxied or extended to operate across a whole homenet. In this context, that may mean bridging a link-local method, taking care to avoid loops, or extending the scope of multicast traffic used for the purpose. It may mean that some proxy or hybrid service is utilised, perhaps co-resident on the CER. Or it may be that a new approach is preferable, e.g. flooding information around the homenet as attributes within the routing protocol (which could allow per-prefix configuration). However, we should prefer approaches that are backwardly compatible, and allow current implementations to continue to be used. Note that this document does not mandate a particular solution, rather it expresses the principles that should be used for a homenet naming and service discovery environment.

One of the primary challenges facing service discovery today is lack of interoperability due to the ever increasing number of service discovery protocols available. While it is conceivable for consumer devices to support multiple discovery protocols, this is clearly not the most efficient use of network and computational resources. One goal of the homenet architecture should be a path to service discovery protocol interoperability either through a standards based translation scheme, hooks into current protocols to allow some for of

communication among discovery protocols, extensions to support a central service repository in the homenet, or simply convergence towards a unified protocol suite.

3.7.2. Assigning names to devices

Given the large number of devices that may be networked in the future, devices should have a means to generate their own unique names within a homenet, and to detect clashes should they arise, e.g. where a second device of the same type/vendor as an existing device with the same default name is deployed, or where two running network elements with such devices are suddenly joined. It is expected that a device should have a fixed name while within the scope of the homenet.

Users will also want simple ways to (re)name devices, again most likely through an appropriate and intuitive interface that is beyond the scope of this document. Note the name a user assigns to a device may be a label that is stored on the device as an attribute of the device, and may be distinct from the name used in a name service, e.g. 'Study Laser Printer' as opposed to printer2.<somedomain>.

3.7.3. Name spaces

If access to homenet devices is required remotely from anywhere on the Internet, then at least one globally unique name space is required, though the use of multiple name spaces should not be precluded. The name space(s) should be served authoritatively by the homenet, most likely by a server resident on the CER. Such name spaces may be acquired by the user or provided/generated by their ISP or an alternative cloud-based service. It is likely that the default case is that a homenet will use a global domain provided by the ISP, but advanced users wishing to use a name space that is independent of their provider in the longer term should be able to acquire and use their own domain name. For users wanting to use their own independent domain names, such services are already available.

Devices may also be assigned different names in different name spaces, e.g. by third parties who may manage systems or devices in the homenet on behalf of the resident(s). Remote management of the homenet is out of scope of this document.

If however a global name space is not available, the homenet will need to pick and use a local name space which would only have meaning within the local homenet (i.e. it would not be used for remote access to the homenet). The .local name space currently has a special meaning for certain existing protocols which have link-local scope, and is thus not appropriate for multi-subnet home networks. A

different name space is thus required for the homenet.

One approach for picking a local name space is to use an Ambiguous Local Qualified Domain Name (ALQDN) space, such as .sitelocal (or an appropriate name reserved for the purpose). While this is a simple approach, there is the potential in principle for devices that are bookmarked somehow by name by an application in one homenet to be confused with a device with the same name in another homenet. In practice however the underlying service discovery protocols should be capable of handling moving to a network where a new device is using the same name as a device used previously in another homenet.

An alternative approach for a local name space would be to use a Unique Locally Qualified Domain Name (ULQDN) space such as .<UniqueString>.sitelocal. The <UniqueString> could be generated in a variety of ways, one potentially being based on the local /48 ULA prefix being used across the homenet. Such a <UniqueString> should survive a cold restart, i.e. be consistent after a network power-down, or, if a value is not set on startup, the CER or device running the name service should generate a default value. It would be desirable for the homenet user to be able to override the <UniqueString> with a value of their choice, but that would increase the likelihood of a name conflict.

In the (likely) event that the homenet is accessible from outside the homenet (using the global name space), it is vital that the homenet name space follow the rules and conventions of the global name space. In this mode of operation, names in the homenet (including those automatically generated by devices) must be usable as labels in the global name space. [RFC5890] describes considerations for Internationalizing Domain Names in Applications (IDNA).

Also, with the introduction of new 'dotless' top level domains, there is also potential for ambiguity between, for example, a local host called 'computer' and (if it is registered) a .computer gTLD. Thus qualified names should always be used, whether these are exposed to the user or not.

There may be use cases where either different name spaces may be desired for different realms in the homenet, or for segmentation of a single name space within the homenet. Thus hierarchical name space management is likely to be required. There should also be nothing to prevent individual device(s) being independently registered in external name spaces.

Where a user is in a remote network wishing to access devices in their home network, there may be a requirement to consider the domain search order presented where multiple associated name spaces exist.

This also implies that a domain discovery function is desirable.

It may be the case that not all devices in the homenet are made available by name via an Internet name space, and that a 'split view' is preferred for certain devices.

This document makes no assumption about the presence or omission of a reverse lookup service. There is an argument that it may be useful for presenting logging information to users with meaningful device names rather than literal addresses.

3.7.4. The homenet name service

The homenet name service should support both lookups and discovery. A lookup would operate via a direct query to a known service, while discovery may use multicast messages or a service where applications register in order to be found.

It is highly desirable that the homenet name service must at the very least co-exist with the Internet name service. There should also be a bias towards proven, existing solutions. The strong implication is thus that the homenet service is DNS-based, or DNS-compatible. There are naming protocols that are designed to be configured and operate Internet-wide, like unicast-based DNS, but also protocols that are designed for zero-configuration local environments, like mDNS [RFC6762].

When DNS is used as the homenet name service, it includes both a resolving service and an authoritative service. The authoritative service hosts the homenet related zone. One approach when provisioning such a name service, which is designed to facilitate name resolution from the global Internet, is to run an authoritative name service on the CER and a secondary resolving name service provided by the ISP or perhaps a cloud-based third party.

Where zeroconf name services are used, it is desirable that these can also coexist with the Internet name service. In particular, where the homenet is using a global name space, it is desirable that devices have the ability, where desired, to add entries to that name space. There should also be a mechanism for such entries to be removed or expired from the global name space.

To protect against attacks such as cache poisoning, it is desirable to support appropriate name service security methods, including DNSSEC.

Finally, the impact of a change in CER must be considered. It would be desirable to retain any relevant state (configuration) that was

held in the old CER. This might imply that state information should be distributed in the homenet, to be recoverable by/to the new CER, or to the homenet's ISP or a third party cloud-based service by some means.

3.7.5. Independent operation

Name resolution and service discovery for reachable devices must continue to function if the local network is disconnected from the global Internet, e.g. a local media server should still be available even if the Internet link is down for an extended period. This implies the local network should also be able to perform a complete restart in the absence of external connectivity, and have local naming and service discovery operate correctly.

The approach described above of a local authoritative name service with a cache would allow local operation for sustained ISP outages.

Having an independent local trust anchor is desirable, to support secure exchanges should external connectivity be unavailable.

A change in ISP should not affect local naming and service discovery. However, if the homenet uses a global name space provided by the ISP, then this will obviously have an impact if the user changes their network provider.

3.7.6. Considerations for LLNs

In some parts of the homenet, in particular LLNs or any devices where battery power is used, devices may be sleeping, in which case a proxy for such nodes may be required, that could respond (for example) to multicast service discovery requests. Those same devices or parts of the network may have less capacity for multicast traffic that may be flooded from other parts of the network. In general, message utilisation should be efficient considering the network technologies and constrained devices that the service may need to operate over.

There are efforts underway to determine naming and discovery solutions for use by the Constrained Application Protocol (CoAP) in LLN networks. These are outside the scope of this document.

3.7.7. DNS resolver discovery

Automatic discovery of a name service to allow client devices in the homenet to resolve external domains on the Internet is required, and such discovery must support clients that may be a number of router hops away from the name service. Similarly the search domains for local FQDN-derived zones should be included.

3.7.8. Devices roaming from the homenet

It is likely that some devices which have registered names within the homenet Internet name space and that are mobile will attach to the Internet at other locations and acquire an IP address at those locations. In such cases it is desirable that devices may be accessed by the same name as is used in the home network.

Solutions to this problem are not discussed in this document. They may include use of Mobile IPv6 or Dynamic DNS, either of which would put additional requirements on to the homenet, or establishment of a (VPN) tunnel to a server in the home network.

3.8. Other Considerations

This section discusses two other considerations for home networking that the architecture should not preclude, but that this text is neutral towards.

3.8.1. Quality of Service

Support for QoS in a multi-service homenet may be a requirement, e.g. for a critical system (perhaps healthcare related), or for differentiation between different types of traffic (file sharing, cloud storage, live streaming, VoIP, etc). Different media types may have different such properties or capabilities.

However, homenet scenarios should require no new QoS protocols. A DiffServ [RFC2475] approach with a small number of predefined traffic classes may generally be sufficient, though at present there is little experience of QoS deployment in home networks. It is likely that QoS, or traffic prioritisation, methods will be required at the CER, and potentially around boundaries between different media types (where for example some traffic may simply not be appropriate for some media, and need to be dropped to avoid drowning the constrained media).

There may also be complementary mechanisms that could be beneficial to application performance and behaviour in the homenet domain, such as ensuring proper buffering algorithms are used as described in [Gettys11].

3.8.2. Operations and Management

The homenet should be self-organising and configuring as far as possible, and thus not be pro-actively managed by the home user. Thus protocols to manage the network are not discussed in this architecture text.

However, users may be interested in the status of their networks and devices on the network, in which case simplified monitoring mechanisms may be desirable. It may also be the case that an ISP, or a third party, might offer management of the homenet on behalf of a user, in which case management protocols would be required. How such management is done is out of scope of this document; many solutions exist.

3.9. Implementing the Architecture on IPv6

This architecture text encourages re-use of existing protocols. Thus the necessary mechanisms are largely already part of the IPv6 protocol set and common implementations, though there are some exceptions.

For automatic routing, it is expected that solutions can be found based on existing protocols. Some relatively smaller updates are likely to be required, e.g. a new mechanism may be needed in order to turn a selected protocol on by default, a mechanism may be required to automatically assign prefixes to links within the homenet.

Some functionality, if required by the architecture, may need more significant changes or require development of new protocols, e.g. support for multihoming with multiple exit routers would likely require extensions to support source and destination address based routing within the homenet.

Some protocol changes are however required in the architecture, e.g. for name resolution and service discovery, extensions to existing zeroconf link-local name resolution protocols are needed to enable them to work across subnets, within the scope of the home network site.

Some of the hardest problems in developing solutions for home networking IPv6 architectures include discovering the right borders where the 'home' domain ends and the service provider domain begins, deciding whether some of the necessary discovery mechanism extensions should affect only the network infrastructure or also hosts, and the ability to turn on routing, prefix delegation and other functions in a backwards compatible manner.

4. Conclusions

This text defines principles and requirements for a homenet architecture. The principles and requirements documented here should be observed by any future texts describing homenet protocols for routing, prefix management, security, naming or service discovery.

5. References

5.1. Normative References

- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.
- [RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.
- [RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, December 2003.
- [RFC3736] Droms, R., "Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6", RFC 3736, April 2004.
- [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.
- [RFC4864] Van de Velde, G., Hain, T., Droms, R., Carpenter, B., and E. Klein, "Local Network Protection for IPv6", RFC 4864, May 2007.
- [RFC5890] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, August 2010.

5.2. Informative References

- [RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.
- [RFC2475] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and W. Weiss, "An Architecture for Differentiated Services", RFC 2475, December 1998.
- [RFC2775] Carpenter, B., "Internet Transparency", RFC 2775, February 2000.
- [RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, May 2000.

- [RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, January 2001.
- [RFC3646] Droms, R., "DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3646, December 2003.
- [RFC4191] Draves, R. and D. Thaler, "Default Router Preferences and More-Specific Routes", RFC 4191, November 2005.
- [RFC4192] Baker, F., Lear, E., and R. Droms, "Procedures for Renumbering an IPv6 Network without a Flag Day", RFC 4192, September 2005.
- [RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, September 2007.
- [RFC5533] Nordmark, E. and M. Bagnulo, "Shim6: Level 3 Multihoming Shim Protocol for IPv6", RFC 5533, June 2009.
- [RFC5969] Townsley, W. and O. Troan, "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) -- Protocol Specification", RFC 5969, August 2010.
- [RFC6092] Woodyatt, J., "Recommended Simple Security Capabilities in Customer Premises Equipment (CPE) for Providing Residential IPv6 Internet Service", RFC 6092, January 2011.
- [RFC6106] Jeong, J., Park, S., Beloeil, L., and S. Madanapalli, "IPv6 Router Advertisement Options for DNS Configuration", RFC 6106, November 2010.
- [RFC6144] Baker, F., Li, X., Bao, C., and K. Yin, "Framework for IPv4/IPv6 Translation", RFC 6144, April 2011.
- [RFC6145] Li, X., Bao, C., and F. Baker, "IP/ICMP Translation Algorithm", RFC 6145, April 2011.
- [RFC6177] Narten, T., Huston, G., and L. Roberts, "IPv6 Address Assignment to End Sites", BCP 157, RFC 6177, March 2011.
- [RFC6204] Singh, H., Beebee, W., Donley, C., Stark, B., and O. Troan, "Basic Requirements for IPv6 Customer Edge Routers", RFC 6204, April 2011.

- [RFC6296] Wasserman, M. and F. Baker, "IPv6-to-IPv6 Network Prefix Translation", RFC 6296, June 2011.
- [RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", RFC 6333, August 2011.
- [RFC6555] Wing, D. and A. Yourtchenko, "Happy Eyeballs: Success with Dual-Stack Hosts", RFC 6555, April 2012.
- [RFC6724] Thaler, D., Draves, R., Matsumoto, A., and T. Chown, "Default Address Selection for Internet Protocol Version 6 (IPv6)", RFC 6724, September 2012.
- [RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, February 2013.
- [RFC6824] Ford, A., Raiciu, C., Handley, M., and O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses", RFC 6824, January 2013.
- [I-D.ietf-v6ops-ipv6-multihoming-without-ipv6nat]
Troan, O., Miles, D., Matsushima, S., Okimoto, T., and D. Wing, "IPv6 Multihoming without Network Address Translation",
draft-ietf-v6ops-ipv6-multihoming-without-ipv6nat-05 (work in progress), March 2013.
- [I-D.ietf-ospf-ospfv3-autoconfig]
Lindem, A. and J. Arkko, "OSPFv3 Auto-Configuration",
draft-ietf-ospf-ospfv3-autoconfig-02 (work in progress), April 2013.
- [I-D.ietf-pcp-base]
Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)",
draft-ietf-pcp-base-29 (work in progress), November 2012.
- [I-D.ietf-v6ops-6204bis]
Singh, H., Beebe, W., Donley, C., and B. Stark, "Basic Requirements for IPv6 Customer Edge Routers",
draft-ietf-v6ops-6204bis-12 (work in progress), October 2012.
- [Gettys11]
Gettys, J., "Bufferbloat: Dark Buffers in the Internet", March 2011,
<<http://www.ietf.org/proceedings/80/slides/tsvarea-1.pdf>>.

[IGD-2] UPnP Gateway Committee, "Internet Gateway Device (IGD) V 2.0", September 2010, <<http://upnp.org/specs/gw/UPnP-gw-WANIPConnection-v2-Service.pdf>>.

Appendix A. Acknowledgments

The authors would like to thank Aamer Akhter, Mikael Abrahamsson, Mark Andrews, Dmitry Anipko, Ran Atkinson, Fred Baker, Ray Bellis, Teco Boot, John Brzozowski, Cameron Byrne, Brian Carpenter, Stuart Cheshire, Julius Chroboczek, Lorenzo Colitti, Robert Cragie, Ralph Droms, Lars Eggert, Jim Gettys, Olafur Gudmundsson, Wassim Haddad, Joel M. Halpern, David Harrington, Lee Howard, Ray Hunter, Joel Jaeggli, Heather Kirksey, Ted Lemon, Acee Lindem, Kerry Lynn, Daniel Migault, Erik Nordmark, Michael Richardson, Mattia Rossi, Barbara Stark, Markus Stenberg, Sander Steffann, Don Sturek, Andrew Sullivan, Dave Taht, Dave Thaler, Michael Thomas, Mark Townsley, JP Vasseur, Curtis Villamizar, Dan Wing, Russ White, and James Woodyatt for their comments and contributions within homenet WG meetings and on the WG mailing list. An acknowledgement generally means that person's text made it in to the document, or was helpful in clarifying or reinforcing an aspect of the document. It does not imply that each contributor agrees with every point in the document.

Appendix B. Changes

This section will be removed in the final version of the text.

B.1. Version 09 (after WGLC)

Changes made include:

- o Added note about multicast into or out of site
- o Removed further personal draft references, replaced with covering text
- o Routing functionality text updated to avoid ambiguity
- o Added note that devices away from homenet may tunnel home (via VPN)
- o Added note that homenets more exposed to provider renumbering than with IPv4 and NAT
- o Added note about devices that may be ULA-only until configured to be globally addressable

- o Removed paragraph about broken CERS that do not work with prefixes other than /64
- o Noted no recommendation on methods to convey prefix information is made in this text
- o Stated that this text does not recommend how to form largest possible subnets
- o Added text about homenet evolution and handling disparate media types
- o Rephrased NAT/firewall text on marginal effectiveness
- o Emphasised that multihoming may be to any number of ISPs

B.2. Version 08

Changes made include:

- o Various clarifications made in response to list comments
- o Added note on ULAs with IPv4, where no GUAs in use
- o Added note on naming and internationalisation (IDNA)
- o Added note on trust relationships when adding devices
- o Added note for MPTCP
- o Added various naming and SD notes
- o Added various notes on delegated ISP prefixes

B.3. Version 07

Changes made include:

- o Removed reference to NPTv6 in section 3.2.4. Instead now say it has an architectural cost to use in the earlier section, and thus it is not recommended for use in the homenet architecture.
- o Removed 'proxy or extend?' section. Included shorter text in main body, without mandating either approach for service discovery.
- o Made it clearer that ULAs are expected to be used alongside globals.

- o Removed reference to 'advanced security' as described in draft-vyncke-advanced-ipv6-security.
- o Balanced the text between ULQDN and ALQDN.
- o Clarify text does not assume default deny or allow on CER, but that either mode may be enabled.
- o Removed ULA-C reference for 'simple' addresses. Instead only suggested service discovery to find such devices.
- o Reiterated that single/multiple CER models to be supported for multihoming.
- o Reordered section 3.3 to improve flow.
- o Added recommendation that homenet is not allocated less than /60, and a /56 is preferable.
- o Tidied up first few intro sections.
- o Other minor edits from list feedback.

B.4. Version 06

Changes made include:

- o Stated that unmanaged goal is 'as far as possible'.
- o Added note about multiple /48 ULAs potentially being in use.
- o Minor edits from list feedback.

B.5. Version 05

Changes made include:

- o Some significant changes to naming and SD section.
- o Removed some expired drafts.
- o Added notes about issues caused by ISP only delegating a /64.
- o Recommended against using prefixes longer than /64.
- o Suggested CER asks for /48 by DHCP-PD, even if it only receives less.

- o Added note about DS-Lite but emphasised transition is out of scope.
- o Added text about multicast routing.

B.6. Version 04

Changes made include:

- o Moved border section from IPv6 differences to principles section.
- o Restructured principles into areas.
- o Added summary of naming and service discovery discussion from WG list.

B.7. Version 03

Changes made include:

- o Various improvements to the readability.
- o Removed bullet lists of requirements, as requested by chair.
- o Noted 6204bis has replaced advanced-cpe draft.
- o Clarified the topology examples are just that.
- o Emphasised we are not targetting walled gardens, but they should not be precluded.
- o Also changed text about requiring support for walled gardens.
- o Noted that avoiding falling foul of ingress filtering when multihomed is desirable.
- o Improved text about realms, detecting borders and policies at borders.
- o Stated this text makes no recommendation about default security model.
- o Added some text about failure modes for users plugging things arbitrarily.
- o Expanded naming and service discovery text.

- o Added more text about ULAs.
- o Removed reference to version 1 on chair feedback.
- o Stated that NPTv6 adds architectural cost but is not a homenet matter if deployed at the CER. This text only considers the internal homenet.
- o Noted multihoming is supported.
- o Noted routers may not be separate devices, they may be embedded in devices.
- o Clarified simple and advanced security some more, and RFC 4864 and 6092.
- o Stated that there should be just one secret key, if any are used at all.
- o For multihoming, support multiple CERs but note that routing to the correct CER to avoid ISP filtering may not be optimal within the homenet.
- o Added some ISPs renumber due to privacy laws.
- o Removed extra repeated references to Simple Security.
- o Removed some solution creep on RIOS/RAs.
- o Load-balancing scenario added as to be supported.

B.8. Version 02

Changes made include:

- o Made the IPv6 implications section briefer.
- o Changed Network Models section to describe properties of the homenet with illustrative examples, rather than implying the number of models was fixed to the six shown in 01.
- o Text to state multihoming support focused on single CER model. Multiple CER support is desirable, but not required.
- o Stated that NPTv6 not supported.
- o Added considerations section for operations and management.

- o Added bullet point principles/requirements to Section 3.4.
- o Changed IPv6 solutions must not adversely affect IPv4 to should not.
- o End-to-end section expanded to talk about "Simple Security" and borders.
- o Extended text on naming and service discovery.
- o Added reference to RFC 2775, RFC 6177.
- o Added reference to the new xmDNS draft.
- o Added naming/SD requirements from Ralph Droms.

Authors' Addresses

Tim Chown (editor)
University of Southampton
Highfield
Southampton, Hampshire SO17 1BJ
United Kingdom

Email: tjc@ecs.soton.ac.uk

Jari Arkko
Ericsson
Jorvas 02420
Finland

Email: jari.arkko@piuha.net

Anders Brandt
Sigma Designs
Emdrupvej 26A, 1
Copenhagen DK-2100
Denmark

Email: abr@sdesigns.dk

Ole Troan
Cisco Systems, Inc.
Drammensveien 145A
Oslo N-0212
Norway

Email: ot@cisco.com

Jason Weil
Time Warner Cable
13820 Sunrise Valley Drive
Herndon, VA 20171
USA

Email: jason.weil@twcable.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 5, 2014

V. Kuarsingh, Ed.
Rogers Communications
J. Brzozowski
Comcast Cable Communications
C. Grundemann
CableLabs
J. McQueen
Broadcom Corporation
July 4, 2013

An incremental solution to advanced home networking
draft-jvkjjmb-home-networking-incremental-00

Abstract

The home network is an environment subject to ongoing evolution and change. Many home networks today are simplistic in nature, often comprising of a single router/gateway. The expectation over time is predicated on the notion that the home network will be more complex servicing many in-home and Internet functions. The home network will evolve necessitating the replacement and update to current hardware and software to more advanced devices and software capable of operating in more complex environments. This document provides a view on how the home network can progress from today's foundational form, to a more advanced environment, using progressive technological capabilities.

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 January 5, 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. Introduction	4
1.1. Requirements Language	5
2. Terminology	5
3. Home Network Progression and Dynamics	6
3.1. Early Home Networks	6
3.2. Home Network Upgrades and Evolution	7
3.3. Home Networking Progression Considerations	7
3.4. Described Phases	9
4. Phase 1: Elementary Network	9
4.1. Service Potential	9
4.2. Topology	10
4.3. Addressing	10
4.4. Routing	11
5. Phase 2: Medius Network	11
5.1. Service Potential	11
5.2. Topology	12
5.3. Addressing	13
5.4. Routing	14
6. Phase 3: Provectus Network	14
6.1. Service Service Potential	15
6.2. Topology	15
6.3. Addressing	15
6.4. Routing	16
7. Phase 4: Posterus Network	18
7.1. Service Potential	18
7.2. Topology	19
7.3. Addressing	20
7.4. Routing	20
8. IANA Considerations	21
9. Security Considerations	21
10. Acknowledgements	21
11. References	21
11.1. Normative References	21
11.2. Informative References	21
Authors' Addresses	23

1. Introduction

The progression of the home IP network to more advanced states will require an evolution from more primitive topologies and protocol support to the eventual advanced topologies with more robust protocol support. Home networks continue to advance from environments originally intended to connect multiple subscriber hosts to a common Internet connection to future environments where Internet, teleworking, home automation, and other functions become common.

In support of this evolution, this document outlines an incremental approach which begins with basic functionality laid out in [RFC6204] and culminates with more advanced topologies and functions. The more advanced home network would align well with the homenet architecture as described in [draft-ietf-homenet-arch] along with potential supporting technologies and concepts like Source Address Routing [draft-troan-homenet-sadr], multi-homing [draft-haddad-homenet-multihomed], IGP based prefix assignment [draft-arkko-homenet-prefix-assignment] and/or [draft-baker-homenet-prefix-assignment].

The criticality and evolution of the customer premises or home network is growing in complexity and importance as the variety and quantity of services being delivered using the Internet Protocol continues to increase. Coupled with these growing needs and the deployment of IPv6 an opportunity exists to advance the state of home networking in an incremental fashion. The introduction of IPv6 support within the home today to facilitate the transition allows for basic Internet access over IPv6, however, this is simply inadequate long term. Home networking technology must advance beyond providing access to Internet content, it must evolve with the goal to improve the customer experience and ensure that the in home network infrastructure is robust and reliable enough to support next generation services.

As part of the evolution and the introduction of IPv6 support the home network support for IPv4 must not be overlooked. Support for IPv4 will remain in devices for years to come and IPv4 will likely continue to be used within the home as IPv4 connectivity to the Internet undergoes dramatic change during the transition to IPv6. As such it is essential that efforts to modernize home networking not only account for both IPv4 and IPv6, these activities must also allow for an incremental approach that does not require flash upgrades of the home networking infrastructure and technology or hardware replacement.

Finally, as home networking technology continues to advance long term the intermediate phases must strive for interoperability to help

ensure a seamless transition and to act as an enabler. This is particular importance for brownfield adopters.

1.1. Requirements Language

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

2. Terminology

Home IP Network Router	a node intended for home or small-office use that forwards packets not explicitly addressed to itself.
Customer Edge Router (CER)	a router that connects the end-user network to a service provider network.
Internal Router	an additional router deployed in the home or small-office network that is not attached to a service provider network. Note that this is a functional role; it is expected that there will not be a difference in hardware or software between a CER and IR, except in such cases when a CER has a dedicated non-Ethernet WAN interface (e.g. DSL/cable/ LTE modem) that would preclude it from operating as an IR.
Down Interface	a router's attachment to a link in the end-user network on which it distributes addresses and/or prefixes. Examples are Ethernet (simple or bridged), 802.11 wireless, or other LAN technologies. A router may have one or more network-layer down interfaces.
Service Provider	an entity that provides access to the Internet. In this document, a service provider specifically offers Internet access using IPv6, and may also offer IPv4 Internet access. The service provider can provide such access over a variety of different transport methods such as DSL, cable, wireless, and others.

Up Interface	a router's attachment to a link where it receives one or more IP addresses and/or prefixes. This is also the link to which a CER or IR points its default route.
depth	the number of layers of routers in a network. A single router network would have a depth of 1, while a router behind a router behind a router would have a depth of 3.
width	The number of routers that can be directly subtended to an upstream router. A network with three directly attached routers behind the CER would have a width of 3.

3. Home Network Progression and Dynamics

3.1. Early Home Networks

Early home networks, as those observed as of the writing of this document, tend to be comprised of few routing zones and layer 3 boundaries. Most home networks attached to operator networks are comprised of a single home LAN segment, or may often have a guest network based on default capabilities of off the shelf vendor platforms.

The object of early home network environments was closely related to providing basic Internet connectivity for subscribers. This trend started back in the late 90's and early 2000s when traditional dial-up connections were replaced with more robust broadband connections. These newer and faster connections provided the bandwidth and flexibility to power home networks which typically used a single gateway towards the Operator's network and Internet.

From an IPv6 perspective this approach was also adopted to ensure that the transition from the use of IPv4 to IPv6 could begin. The objective here has been to enable as long a period of time as possible to encourage the incremental transition to the use of IPv6 while avoiding or delaying the use impactful and costly transition technologies.

Subscriber expectations during the early network phases were to connect multiple machines to the Internet over a single connection. Early Internet services were often traditional web (HTTP), Mail (SMTP/POP3/IMAP) and news (NNTP) among others. Most services other than the Internet were often supported by legacy technologies. Such

other services may have included legacy Video (Cable and/or Satellite TV), Voice (PSTN) and the like.

3.2. Home Network Upgrades and Evolution

The home network, although originally used for very basic Internet connectivity, is now beginning to expand to support many more services. Some of the historical expansion within the home includes the use of the home network for Media like Photo viewing, Movies viewing and communications between in-home devices. During the 2000s, the home network also saw a strong growth on how it was used for more commercial functions such as online video streaming, peer to peer communications, remote teleworking, and social networking.

The expansion to date did not necessarily require more complex home networks, but did expand the use of the common IP connection provided to subscribers. Some historical expansion within the home networks were related to advancement of legacy services such as voice which has transformed to VoIP in recent years. Operators often used technological frameworks, such as Packet Cable [PacketCable], to enable legacy services over IP. These newer functions expanded the home network or at times the gateway functions attaching to the operator's network. Future expansion and/or enhancements similar to VoIP in some operator networks include IPTV.

The evolution of the home network continues as subscribers are beginning to use the home network to connect many new IP enabled devices. These new IP enabled devices include home security endpoints, sensors, appliances, home electronics among many others. As these new uses become prevalent in the home network, so do the needs of the attached devices, and the somewhat arbitrary ecosystem that is used to attach them. It is the expectation that this expanding ecosystem will create more robust and topologically complex home networks. It is not well understood how long or fast this evolutionary path will take, but the evolution has started.

The topological attachment and addressing needs within the home network will need to be supported by the infrastructure which comprises the home network. The upstream operators will need to be cognizant of this need such as to provide the correct address sizes and/or address elements. The underlay routing platforms will also need to support the evolving home network to allow for growth and robust home network environment.

3.3. Home Networking Progression Considerations

Evolving from basic to intermediate or advanced home networks there are a number of considerations. These considerations range from

technical matters specific to implementation to operational and deployment considerations.

Unlike green field deployments most operators and customers must take into consideration that existing products and services must continue to be supported. It is also important to note that the process to upgrade must be seamless and non-disruptive. Disruption during the upgrade process will not only have direct impact to the customers of the same, but also directly impact the operators who deliver the same. Support call volumes and impacts to operator infrastructure must be factored in to the process of new technology introduction whether it is basic support for IPv6 Internet connectivity or intermediate/advanced home networking.

While most modern home network devices are software upgradeable to introduce new functionality, some feature and enhancements exceed device capabilities. Further there is a large population of home networking devices that are based on an earlier generation of technology as such may not be upgradeable to even the most basic form of home networking. Fortunately operators actively work to ensure their infrastructure is modernized to ensure the greatest feature set and performance are available to their customers.

For those platforms that are upgradeable, which is a non trivial population for most operators, the implementation process can be complex. Complexity is presented in many forms ranging from the technical details to integration with other essential features that must be supported. Technical functionality must be balanced with simplicity. Features that are overly complex impact implementers, customers, and operators. As such it is essential that technical solutions be implementable and supportable by vendors, deployable by operators, and usable by customers. Testing and interoperability are critical aspects of advancing any technology; this is especially the case with home networking largely due to the significant quantity of unique devices that are in use with broadband enabled home networks. The quantity and types of devices that are connected today are vast and unique, there are probably very few home networks that are alike today. As such innovators of the home network must ensure that they develop the technologies with support for legacy devices while laying the ground work for the next generation of connected devices and the services that they enable.

Continued support for IPv4 to the Internet is a must at the time this document was written. While efforts advance to someday enable IPv6 only with consumer electronics and content providers support for IPv4 remains essential and cannot be ignored or forgotten. Further, at the time IPv6 only becomes a reality, it is likely that IPv4 (even for dual stack in home devices) will continue to be used, possibly

for decades to come. Since it is difficult to determine if or when IPv4 will no longer be required operators and innovators must ensure that IPv4 continues to be supported.

3.4. Described Phases

The phases described below are intended to provide a descriptive of how the overall transition from early home networks can evolve to advanced home networks that support more complex functions. These phases are labeled as "Elementary" (Initial), "Medius" (Middle), "Provectus" (Advanced) and "Posterus" (Subsequent). The phase labels chosen are for reference purposes only within this document and do not hold any significance for global meaning.

4. Phase 1: Elementary Network

The Elementary network phase is closely associated to the basic environment described in section two of this document. The phase is based on basic connectivity from a simple home network toward the Internet. As a starting point, future phases are built assuming this phase was generally present. While some greenfield environments may emerge, most future more advanced home networks will expand out of a basic home network.

The Elementary network will likely have basic functionality at the operator/Internet edge (subscriber's point of reference) as outlined in [RFC6204]. In addition to IPv4 legacy functionality, [RFC6204] lays out basic requirements for IPv6 connectivity. With reference to IPv6, the main emphasis was the attachment of the home network to the IPv6 Internet, along with any legacy attachment to the IPv4 Internet.

4.1. Service Potential

In the Elementary network, as noted, connectivity is quite basic and allows a simple home network to connect to the IPv4 and/or IPv6 Internet. Services envisioned to be used in this phase are very similar to those which are already well established. Such services include traditional Internet services such as web (HTTP), mail, news, ftp, peer to peer, social networking, teleworking among many others.

Given the simplistic topological nature of this phase, there is less flexibility for downstream segments to be attached, and the in-home environment that is supported is assumed to be flat in nature (single layer 2 domain, with a potential secondary environment such as a guest net). Attaching networks, like a sensor net may not be well serviced in such a model since such environments may require layer 3 separation and/or advanced device functionality.

4.2. Topology

The topology in the Elementary phase is basic in nature and often assumed to be flat with some exceptions for additional segments like a guest net. The topology in this phase may bridge multiple switched and WiFi environments. It is possible that tandem routers may show up in the Elementary phase, and those environments would topologically be downstream sub-network environments with little integration with the upstream IPv4 environment and likely no support for IPv6 on the sub-network/tandem LAN network (not shown in Figure 1 below).

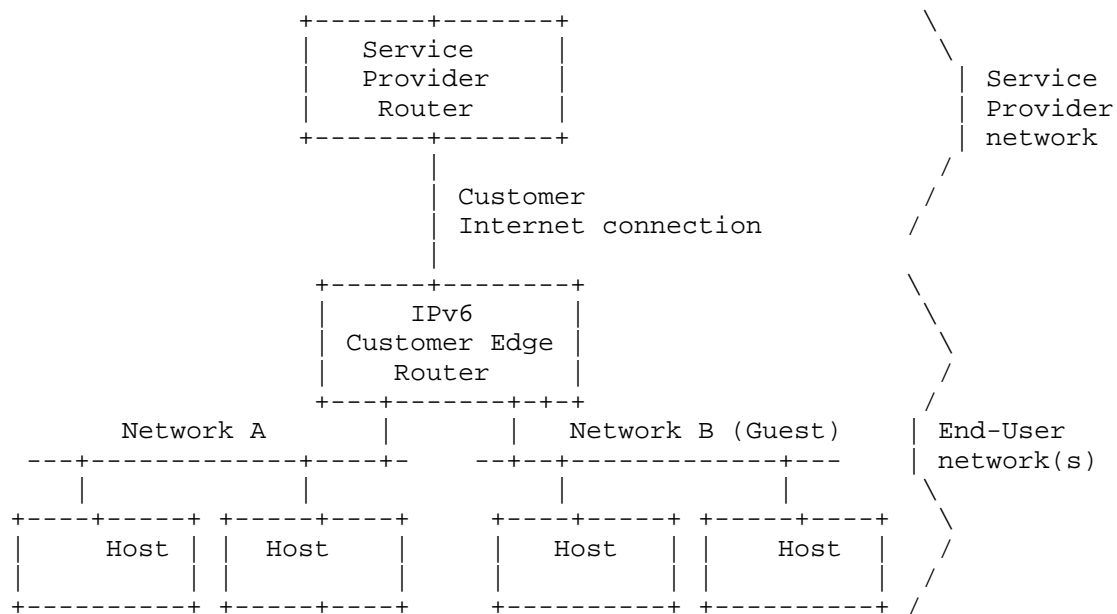


Figure 1: Basic Home Network

4.3. Addressing

Addressing in the Elementary phase will be supported by legacy IPv4 addressing behaviours and IPv6 addressing behaviours as described in [RFC6204]. IPv4 would operate much like it has for many years using [RFC1918] addressing in the home network and translating (NAT44) traffic flows to/from the Internet using a single IPv4 global address assigned to the subscriber's gateway (operating facing interface). IPv6 is expected to utilize a prefix delegation as described in [RFC6204] and allow addresses from within the delegated prefix to be assigned to hosts (or auto-configured by hosts using announced prefix) on the guest network.

For cases where the operator supplies a single /64, only a single segment can be supported on the home network. In cases where larger prefixes are assigned to the subscriber's gateway, additional segments can receive a /64 assignment for use on those segments. For the most part, those segments are assumed to be attached directly off the customer's edge router. Network side addressing will typically be provided by DHCPv6 [RFC3633] or RADIUS [RFC4818].

4.4. Routing

Routing in the Elementary network is also quite basic in nature. Since the layer 3 segments are typically attached to a single gateway, routing is conducted by the single gateway which uses a default route pointing to the upstream provider. The provider will have gleaned routing state which is gleaned from the DHCPv6 transaction and/or RADIUS addressing as stated in the section above.

5. Phase 2: Medius Network

The Medius network extends and builds upon the concepts established in the Elementary phase. The Medius network is intended to incrementally enable and evolve the home network by going beyond basic IPv4 and/or IPv6 access to the Internet. In the Medius phase, the objective are to go beyond basic access to content and services on the Internet. Medius leverages the basic concepts established in the Elementary phase as building blocks for more advanced in home functionality while acting as a foundation for future iterations of home networking. Medius will not only act as an enabler but will also provide criticality required transition capabilities to ensure advanced phases can also be deployed seamlessly to both customers and operators with minimal disruption.

5.1. Service Potential

Medius enables natively routable IP within a home or customer premise. Leveraging mostly existing techniques and technology, Medius fortifies and advances the home network to support the delivery of next generation content and services while maintaining balance from an implementation and deployment point of view. A Medius home network alleviates the notion of nest IPv4 address and port translation within a home or premises so that both IPv4 and IPv6 communications are high performance and reliable. Next generation applications, services, and content will benefit greatly from a native environment within the home as it is well known that address and port translation not only introduce latency and delays but also hamper the premise wide experience for customers. A native environment will allow customers to take full advantage of all

devices and applications that are and will appear throughout their home networks.

5.2. Topology

The topology of a Medius home network, unlike that of the Elementary phase, will be that of a native IP network in that there is expected to be no translation within the premises. The only place where translation is likely to occur is at the premises edge facing the Internet or service provider and will be for IPv4 only, which is common place today for most home networks. Non-Medius devices will continue to be supported, however, the experience and functionality of a Medius environment may not be fully achievable. The Medius network will leverage a tree topology and can support flexibility in how sub-tended devices are connected and provisioned. However, while the Medius topology can support sub-tended devices the expectation here is that the advanced home networking phase will introduce support for greater flexibility as the need arises. The flexibility of the Medius topology meets the near to mid term flexibility requirements for the customer premises. In a Medius home network interface detection is supported allowing for devices to be connected and reconnected to one another in a manner that will enable each to be reconfigured along with any connected devices. Unlike the Elementary phase, Medius will support multiple routed segments and a home network hierarchy not just a primary segment and guest segment providing greater flexibility in how devices within the customer premises connect and access content and services over the internal segments and on the Internet.

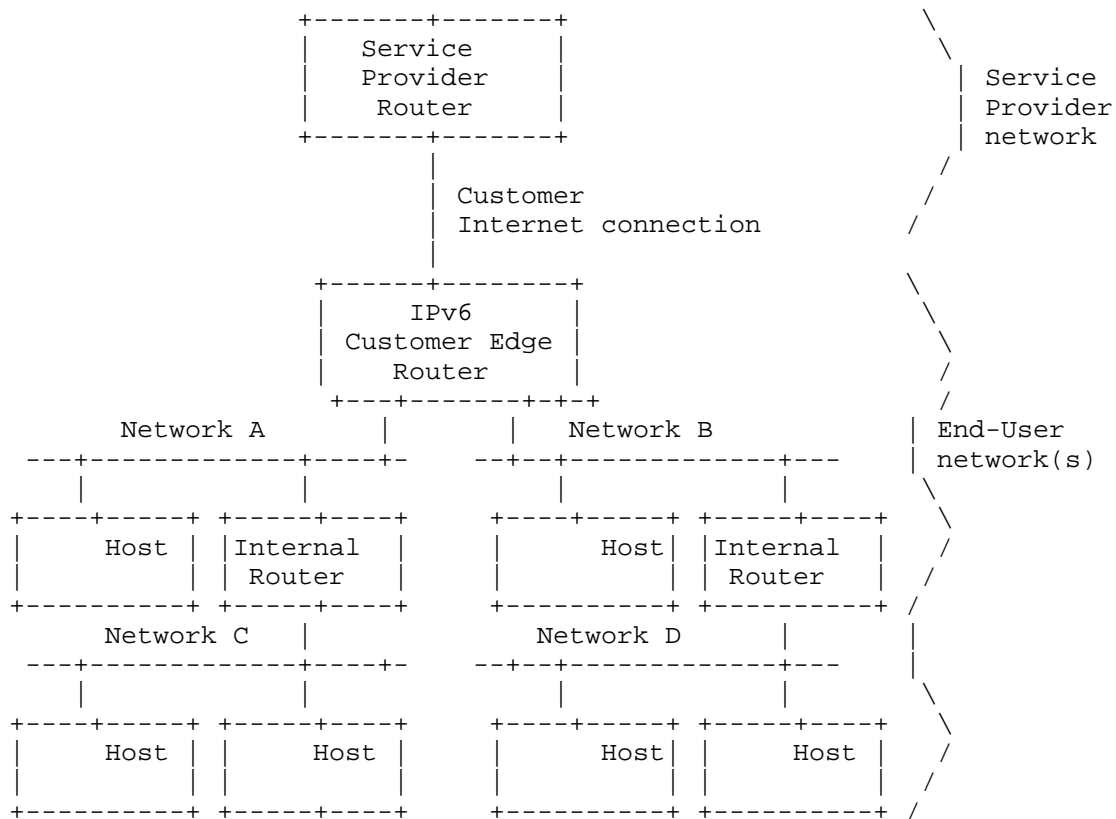


Figure 2: Progressive Home Network

5.3. Addressing

The Medius network assumes greater flexibility and extensibility, and as such, the addressing within the premises must be comparable. Nodes or endpoints will be able to leverage all common techniques for both IPv4 and IPv6 addresses. For IPv4, DHCP and static addressing will be supported, however, dynamic addressing techniques are preferred. For IPv6, stateless auto-configuration with [RFC6106] support, stateless and stateful DHCPv6, and static addressing are all supported. Stateful DHCPv6 includes support for IPv6 prefix delegation [RFC3633]. For IPv6, all forms of dynamic addressing are preferred over any form of static assignments.

Sub-tended routing devices in a Medius home network will leverage IPv6 prefix delegation [RFC3633]. The width and depth of the duo home network will provide the guidance required to determine the size of sub-delegated prefixes to ensure the greatest depth and width for

a Medius home network. While Medius home networks have flexibility for sub-delegated networks, it is expected that the typical Medius home network will have a limited quantity of connected, sub-tended devices. The overarching objective of the Medius network is to alleviate overly flat home networks while enabling high performance, reliably home networks that can be used to provide service to and support next generation content, services, and applications.

In a Medius home network, IPv6 techniques can be leveraged to bootstrap the provisioning and enablement of IPv4 natively. The provisioning and enablement of IPv4 care of IPv6 enables congruent IPv4 and IPv6 topologies and routing within a customer premises. Congruent IPv4 and IPv6 help to ensure that ideal conditions are in place within the home to enable the greatest performance, reliability, and stability by dynamically altering security and routing configurations. The details of IPv6 bootstrapped IPv4 provisioning is out of scope for this document.

5.4. Routing

The Medius home network does not currently require a dynamic routing protocols for routing or provisioning purposes. Provisioning techniques specified for Medius home networks are outlined in the addressing section above. Routing for duo home networks is governed by each routing device. The customer or premise edge router (facing the Internet or service provider) is aware of both IPv4 and IPv6 routing information, care of the provisioning process, for sub-tended device. Each router in a duo home network will act as a default route for all connected devices. If a routing device supports multiple connected, routed segments, routing information for all connected segments will be known inherently by the device, otherwise, the device will send all traffic to its own default route learned from its "up" interface.

The Medius topology, addressing, and routing collectively act as foundational elements for future states of the home network. In fact, advanced home networking techniques may not be achievable without aspects of Mediux being deployed in customer premises networks.

6. Phase 3: Provectus Network

The Provectus phase is a progressive option beyond the Medius network with the option to turn on a routing protocol. Efficiency can be gained here while keeping addressing common from the previous model. This model effectively allows for a routing protocol to be added (will be explained in routing section below in more detail) allowing

for better flow efficiency in-home, likely helping more advanced home networks where there are more in-home services/operations.

A routing protocol can be added to an existing Medius Network as an "add on" feature. Among the advantages of having a routing protocol enabled is that it provides an efficient and dynamic network topology in the home. With respect to efficiency, a routing protocol will aid in maintaining a routing table with the sense of metric or hops to destinations. Additionally, a routing protocol can aid in the network dynamic scalability.

The details of which specific routing protocol to use are out of scope for this text; however, we will discuss how some of the common routing protocols could help improve on an existing Medius Network.

6.1. Service Service Potential

The main service advantage of a Provectus Network is that it will provide more seamless flexibility and transition when routers in the home are either added, removed, or moved to a different location in the network. Additionally, a routing protocol can provide a more seamless transition upon IP configuration changes. It is important to note that the added flexibility and efficiency may be lost when there are inline routers which do not support or enable the same routing protocol. In such a case, the system may revert to the Medius network operation (not as efficient, but functional).

6.2. Topology

The topology for the Provectus network phase is expected to be similar to that of the Medius network phase.

6.3. Addressing

The Provectus Network IP addressing considerations match those of the Medius phase with all IP addressing adhering to the HIPNet DHCPv6 and recursive prefix delegation model described above.

All downstream routers to the CER will request an IPv6 prefix (IA_PD) via DHCPv6 along with an IPv6 address (IA_NA). DHCPv6 will provide address and prefix information for a specific lifetime.

Upstream routers are able to create a routing table based on the address and prefix information provided to downstream routers while the DHCP lease is active. Without a routing protocol, upstream routers will not be instantly aware of when a downstream router is being removed from or moved within the home network. The upstream routers will eventually learn of a removed or moved router when the

DHCPv6 lifetime expires or if the router imposes other gleaning or keep alive logic to track router movement.

In another scenario, an upstream HIPnet router's width or depth could be presently maximized and there is the need to replace one router connected on the width or depth with another router. The upstream router will be unable to provision the new router on the network until it detects that the previous router has been removed from the network. Without a routing protocol, the time needed to discover this topology change will most likely be longer and it places a dependency on the router which is not implementing a routing protocol to assure that it cleans up its own routing table when DHCP leases expire.

6.4. Routing

A routing protocol provides either instant and/or periodic notification of the addition and deletion of downstream routers.

HIPNet-based upstream routers are aware of the prefixes that are provisioned to the downstream routers and on which interface that prefix route is associated. The downstream routers will refresh and/or update that initial routing configuration.

A routing protocol simplifies the efforts needed by HIPNet routers not implementing a routing protocol by eliminating the probing of routing information with DHCPv6 and perhaps, Neighbor Discovery [RFC4861] messages of downstream routers.

Additionally, a routing protocol could help support a router with manual IPv6 prefix configuration. In a HIPNet router configuration, recursive prefix delegation is assumed to cascade prefix information to sub-tending downstream routers. However, with a routing protocol, it could support a directly connected router or line of routers with manual prefix configuration.

The routing protocol detailed here is assumed to be carrying only route information and does not consider the presence of any other configuration data.

Initially, it should also be assumed that the same routing protocol is used within the home network and there would not be the need for redistributing between multiple routing protocols.

Additionally, it is assumed that there will be no configuration information passed along within the routing protocol for the Provectus phase. That is, the routing protocol will simply be used to maintain and update routing tables.

Some standard routing protocols to be considered for a Provectus network (but not limited to) are:

- RIPng [RFC2080]
- OSPFv2 [RFC2328] and/or OSPFv3 [RFC5340] / [RFC5838]
- IS-IS (may also be candidate for phase 4 with multihoming) [ISO-ISIS]

RIPng

- Distance vector protocol (hop count based)
- UDP protocol, port 521
- Multicast based messaging
- Authentication done by IPv6 IPSEC layer

Advantages of RIPng

- Simple in design/implementation
- Supported a network topology change immediate route update

Disadvantages of RIPng

- Naturally converges slowly with default 30 second reporting interval
- Multicast advertisement of complete routing table on every reporting interval
- Hop Limit max of 15, 16 means infinity and considered unreachable
- Requires split horizon to report only those routes not sourced by the destination router.

OSPF

- Link state protocol (route cost based)
- IP based protocol
- Authentication done by IPv6 IPSEC layer

Advantages of OSPF

- Only updates when change in route table occurs - low network overhead
- Limited only by size of routing table
- Low convergence time

Disadvantages of OSPF

- More complex in design/implementation
- May be difficult to configure

7. Phase 4: Posterus Network

The Posterus phase of home network development looks beyond many of the current home networking paradigms and allows more dramatic changes to surface. This phase can be characterized by two general enhancements over previous phases; IGP enhancements allowing things like prefix distribution and better multihoming capabilities, possibly through source address route selection as described in documents like [draft-troan-homenet-sadr] . The Posterus phase is the focus of much of the current work of the Homenet working group.

Because the Posterus phase is the furthest into the future, it has the most potential to change over time. As such, this document captures current ideas and thinking but should not be seen as limiting in any way the potential for future progress within home networks. Additionally, many of the more dramatic changes currently envisioned for this phase (e.g. IGP prefix distribution) are not backwards compatible with existing home routers, nor those that are likely to emerge in the Medius and Provectus phases. This fundamental principle may restrain the Posterus phase deployments to those home networks which actively require the added functionality and efficiency.

7.1. Service Potential

The primary and most obvious service enhancement of the Posterus phase is the enhancement of multihoming, connecting the home network to multiple ISPs simultaneously for added bandwidth and/or enhanced resiliency to WAN connectivity outages. Of course, multihoming isn't limited only to multiple ISP connections, it could also enable the connection of multiple discreet home networks thus enabling new services related to local content sharing and caching. Also, as this

phase leverages routing protocols specifically tailored to home networking, further service enhancements may be possible. Service discovery is one area which may benefit from such enhancements if these routing protocols are extended to share such information.

7.2. Topology

Posterus supports longer term, advanced home network topologies. As mentioned above, this is primarily focused on enabling multiple ISP connections, which is illustrated in figure 3, below. Other topological changes in this phase may include connections to other, non-ISP networks and other more complex home network topologies.

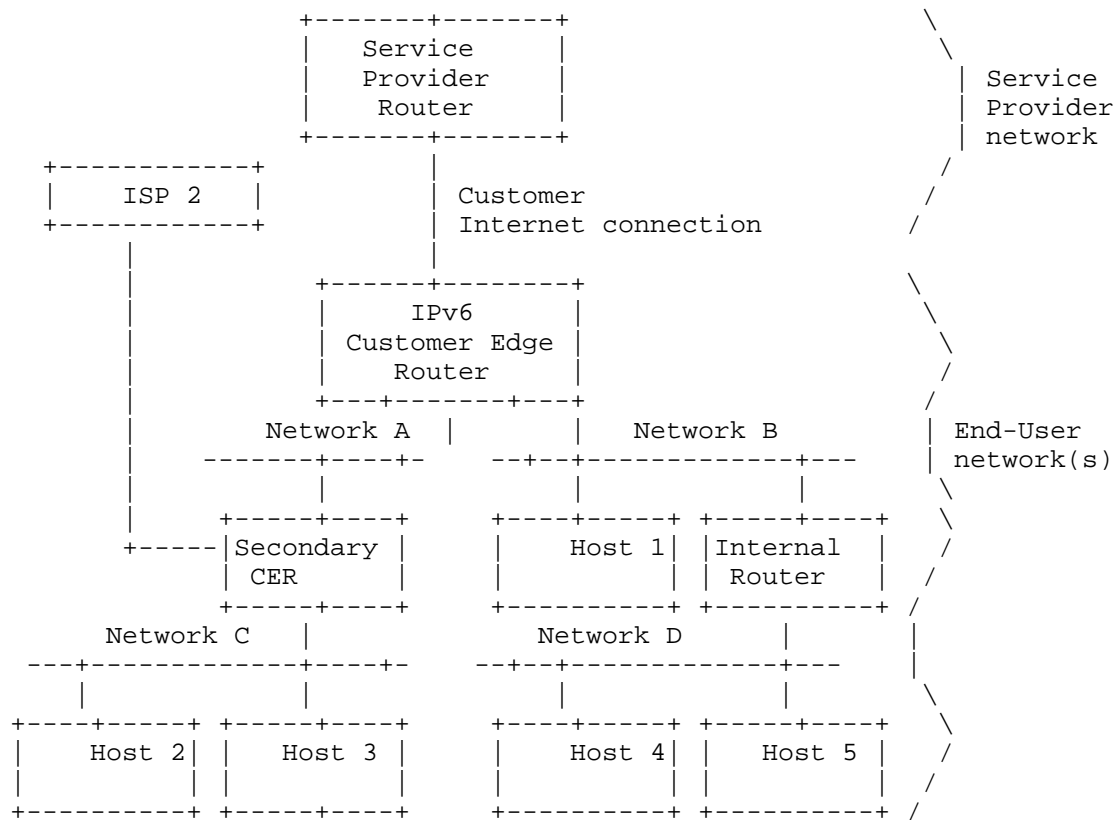


Figure 3: Complex Network with Two ISP Connections

7.3. Addressing

Posterius has the potential to see the most dramatic shift in addressing of all the phases documented here. As described in [draft-arkko-homenet-prefix-assignment], one of the key enhancements in this phase may be the distribution of prefix information through an enhanced IGP. The use of IGP prefix distribution has the distinct advantage of breaking from the hierarchical prefix distribution mechanisms introduced in the Medius phase, enabling much more efficient use of the prefix' available to the home network. This is a significant advantage in home networks that are allocated a small prefix, such as a /60. Also, while the "Link ID" concept introduced in the Medius phase will allow for the use of multiple address families and multiple distinct prefixes within the home network, the IGP prefix distribution also promises to further facilitate the use of prefix' from more than one ISP.

The inherent complication introduced by IGP prefix distribution is one of backwards compatibility. Home routers in the Elementary, Medius and Provectus phase, in addition to all current home routers, use DHCPv6 PD [RFC3633] exclusively for prefix distribution within the home. A home router designed to use DHCP [RFC3633] will be able to provide a prefix to a downstream Posterius phase router, but would be unable to accept a prefix delegated from an upstream Posterius phase router via an enhanced IGP.

7.4. Routing

Routing in Posterius will likely introduce source address route selection, currently described in [draft-troan-homenet-sadr] and [draft-xu-rtgwg-twod-ip-routing]. Source address route selection uses both the source and destination addresses when determining the proper routing of packets leaving the home network. This will greatly enhance the multihoming capabilities of home networks by ensuring that communications initiated within the home network will always use the proper upstream ISP, avoiding ingress filtering present on most residential broadband access networks like [RFC2827].

In addition to enhanced multihoming capabilities, source address route selection may also be leveraged for access control within home networks. Since each layer 3 domain within a home network can be identified by the specific prefix' being used on that segment, source address based access control provides an effective means of implementing policy in routing.

8. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

9. Security Considerations

No security considerations noted at this time.

10. Acknowledgements

Special thanks for the following people for their contributions.

11. References

11.1. Normative References

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

11.2. Informative References

[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.baker-homenet-prefix-assignment]
Baker, F. and R. Droms, "IPv6 Prefix Assignment in Small Networks", draft-baker-homenet-prefix-assignment-01 (work in progress), March 2012.

[I-D.haddad-homenet-multihomed]
Haddad, W. and D. Saucez, "Multihoming in Homenet",
draft-haddad-homenet-multihomed-01 (work in progress),
March 2013.

[I-D.ietf-homenet-arch]
Chown, T., Arkko, J., Brandt, A., Troan, O., and J. Weil,
"Home Networking Architecture for IPv6",
draft-ietf-homenet-arch-08 (work in progress), May 2013.

- [I-D.troan-homenet-sadr]
Troan, O. and L. Colitti, "IPv6 Multihoming with Source Address Dependent Routing (SADR)",
draft-troan-homenet-sadr-00 (work in progress),
February 2013.
- [I-D.xu-rtgwg-twod-ip-routing]
Xu, M., Wu, J., Yang, S., and D. Wang, "Two Dimensional IP Routing Architecture", draft-xu-rtgwg-twod-ip-routing-00
(work in progress), March 2012.
- [ISO-ISIS]
"ISO/IEC 10589:2002 Intermediate System to Intermediate System intra-domain routing information exchange protocol", 3 2008, <http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=30932>.
- [PacketCable]
CableLabs, "Packet Cable™ 2.0 Architecture Framework Technical Report", May 2009, <<http://www.cablelabs.com/specifications/PKT-TR-ARCH-FRM-V06-090528.pdf>>.
- [RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets",
BCP 5, RFC 1918, February 1996.
- [RFC2080] Malkin, G. and R. Minnear, "RIPng for IPv6", RFC 2080,
January 1997.
- [RFC2131] Droms, R., "Dynamic Host Configuration Protocol",
RFC 2131, March 1997.
- [RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, April 1998.
- [RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, May 2000.
- [RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633,
December 2003.
- [RFC4818] Salowey, J. and R. Droms, "RADIUS Delegated-IPv6-Prefix Attribute", RFC 4818, April 2007.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861,
September 2007.

- [RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, July 2008.
- [RFC5838] Lindem, A., Mirtorabi, S., Roy, A., Barnes, M., and R. Aggarwal, "Support of Address Families in OSPFv3", RFC 5838, April 2010.
- [RFC6106] Jeong, J., Park, S., Beloeil, L., and S. Madanapalli, "IPv6 Router Advertisement Options for DNS Configuration", RFC 6106, November 2010.
- [RFC6204] Singh, H., Beebee, W., Donley, C., Stark, B., and O. Troan, "Basic Requirements for IPv6 Customer Edge Routers", RFC 6204, April 2011.

Authors' Addresses

Victor Kuarsingh (editor)
Rogers Communications
8200 Dixie Road
Brampton, ON L6T 0C1
Canada

Email: victor@jvknet.com

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

Email: john_brzozowski@cable.comcast.com

Chris Grundemann
CableLabs
858 Coal Creek Circle
Louisville, CO 80027
USA

Email: c.grundemann@cablelabs.com

John McQueen
Broadcom Corporation
16340 West Bernardo Drive
San Diego, CA 92127
USA

Email: jmcqueen@broadcom.com

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: January 16, 2014

M. Le Pape
S. Bhandari
Cisco Systems
I. Farrer
Deutsche Telekom AG
July 15, 2013

IPv6 Prefix Meta-data and Usage
draft-lepape-6man-prefix-metadata-00

Abstract

This document presents a method for applications to influence the IPv6 source selection algorithm used by the IP stack in a host. To do so, IPv6 prefixes are associated with meta-data when configured by the network. This meta-data allows the network to describe the purpose and properties of the prefix enabling applications to make intelligent decision when selecting a prefix.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC2119 [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 January 16, 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. Introduction	3
1.1. Motivation	3
1.1.1. Home networks	3
1.1.2. Mobile networks	4
2. Overview	5
3. Considerations	7
3.1. Prefix meta-data propagation	7
3.2. Configuring Applications	7
3.3. Application to network stack communication	8
3.4. Default Address Selection	8
3.5. Scope of Prefix Color	8
3.5.1. Local scoping	9
3.5.2. Local scoping with fuzzy matching	9
3.5.3. Global scoping	9
3.6. Compatibility with Existing Implementations	9
4. IANA Considerations	10
5. Security Considerations	10
6. Acknowledgements	10
7. Change History (to be removed prior to publication as an RFC)	10
8. References	10
8.1. Normative References	10
8.2. Informative References	10
Appendix A. Prototype notes	12
A.1. Homenet prototype implementation notes	12
A.1.1. Video provider service	12
A.1.2. Prefix Color delegation	12
A.1.3. Configuring Applications	13
A.1.4. Android DHCPv6	14
A.1.5. Application to network stack communication	14
A.1.6. Android kernel	15
A.1.7. Limitations of the current prototype	16

Authors' Addresses	16
--------------------	----

1. Introduction

IPv6 provides not only a larger address space than IPv4, but also allows host interfaces to have more than one IPv6 address of the same or different scope(s). When multiple prefixes are assigned to one or more network interfaces each of the prefixes can have a specific property and purpose associated with it. For example: In a mobile network, a mobile device can be assigned a prefix from its home network and another from the visiting network that it is currently attached to. Another example is a public WLAN hotspot configured with two prefixes offering Internet access. One is free, but low-quality, whilst the other is charged and offers service level guarantees.

A prefix may have well defined properties that are universal and have additional meta-data associated with it in order to communicate the prefixes local significance. When multiple prefixes are provisioned to the host, this additional information allows the host and applications to make more intelligent decisions about the best IPv6 address to select when sourcing connections.

This document introduces the motivations and considerations for having additional meta-data associated with a prefix and also proposes a format for the meta-data itself.

The underlying assumption is that a endpoint or an application has multiple prefixes to choose from. Typically this means either the endpoint has multiple interfaces or an interface has been configured with multiple prefixes. This specification does not make a distinction between these alternatives.

1.1. Motivation

In this section, the motivation for attaching meta-data to IPv6 prefixes is described in the context of both mobile and home networks. The meta-data helps to distinguish an IPv6 prefix and aids with the selection of the prefix by different applications.

1.1.1. Home networks

In a fixed network environment, the homenet CPE may also function as both a DHCPv6 client (requesting IA_PD(s)) and a DHCPv6 server allocating prefixes from delegated prefix(es) to downstream home network hosts. Some service providers may wish to delegate multiple globally unique prefixes to the CPE for use by different services classes and traffic types.

Motivations for this include:

- o Using source prefix to identify the service class / traffic type that is being transported. The source prefix may then reliably be used as a parameter for differentiated services or other purposes. E.g. [I-D.jiang-v6ops-semantic-prefix]
- o Using the specific source prefix as a host identifier for other services.
- o In multi-homed environments, a single homenet LAN may have multiple globally unique prefixes provided by the different service providers. In this scenario, correct source address selection is fundamental to successfully establishing connections. E.g. [I-D.troan-homenet-sadr]

Any host which is configured with multiple prefixes must perform a source address selection process when initiating a connection. Any client that has multiple globally unique prefixes only has source and destination longest-prefix matching policy [RFC6724] in order to make this selection. For cases such as those listed above, longest-prefix matching can not assist the client in selecting the correct source address to use. Additional information is needed to assist the client in making the correct source address selection.

1.1.2. Mobile networks

In mobile network architecture, a mobile node can be associated with multiple IPv6 prefixes belonging to different domains. E.g. home address prefix, care of address prefix (as specified in [RFC3775]). The delegated prefixes may be topologically local and some may be remote prefixes anchored on a global anchor, but available to the local anchor by means of tunnel setup in the network between the local and global anchor. Some prefixes may be local with low latency characteristics suitable for voice call break-out, some may have global mobility support.

So, the prefixes have different properties and it is necessary for the application using the prefix to learn about this property in order to use it intelligently. An example is determining if the prefix is a home address or care of address or other network characteristics that can be offered.

2. Overview

The mechanism that is described in this document describes two different types of meta data which can be used in different ways:

Prefix Properties Provides a method for an application to "hint" required source address properties to the kernel. These properties are universal and expressed as a set of flags.

Prefix Color Provides an arbitrary color value to prefixes (of local significance) enabling an application to request a source prefix with a specific color.

These two meta data types are described in more detail below.

Prefix Properties functions as follows:

- o The client receives multiple prefixes, with relevant Prefix Property meta-data attached to each prefix
- o Prefix property aware applications running on the client have a policy defining that they prefer prefixes that have specific properties.
- o On initiating a connection, the Prefix Property aware application passes the required prefix properties to the kernel along with the connect request
- o The kernel checks the requested properties against the available prefixes. If a match is found, the matching prefix is passed back to the application
- o The application uses the returned prefix when making the call to the socket API to create the connection
- o If no prefix matching the requested properties is available, then the kernel uses [RFC6724] for source address selection as normal

Prefix property offers well defined universally understood information about the prefix. Example properties include whether a prefix can provide Internet reachability, if the prefix offers application specific Internet service level, if the prefix usage is free/charged, if the prefix offers security guarantees etc. This is maintained as a global registry.

Prefix Coloring functions as follows:

- o The client receives multiple prefixes, with relevant meta-data attached
- o Color aware applications running on the client are provisioned with policy telling them which prefix color to request
- o On initiating a connection, the meta data aware application passes the required prefix color to the kernel along with the connect request
- o The kernel takes this color and selects the prefix matching the requested color and passes this back to the application
- o The application uses the returned prefix when making the call to the socket API to create the connection

Prefix colour conveys information of the prefix that is of relevance to the network where the prefix is provisioned and application using it. Example usage of prefix color include color that is provisioned to offer better video application experience. The prefix color is defined as a 16 bit numerical value.

Figure 1 illustrates a typical network with different components that can add, understand and use the meta-data attached to a prefix.

- o Mobile or ISP Network - Provisioned with prefixes that offer specific network characteristic. e.g. prefixes that do not have internet reach but can offer quality of service required for better video application experience. Includes address delegation server that associate prefixes with this information, selects and offers this information during prefix delegation
- o Home/Mobile gateway - Learns or determines characteristic of the prefix and propagates it along with prefix delegation. e.g. Determines if the prefix is locally anchored or learns the prefix meta-data from the ISP prefix delegation server and includes this information in prefix delegation to endpoints
- o Endpoint network stack - Learns the additional information associated with the prefix and offers interface to applications for listing and selecting the available prefixes
- o Prefix selection policy - Either embedded in the application/endpoint or learnt from a server that helps choose the prefix with specific characteristic for the application based on predetermined service agreement between the application/endpoint/application service provider and network service provider

- o Applications - That can utilize the prefix with specific characteristic for enhanced application user experience e.g. On demand video application, by choosing the prefix with appropriate prefix selection policy while connecting and delivering the application over the network

This prefix meta-data could be further extended to have more attributes such as the administrative domain of the prefix.

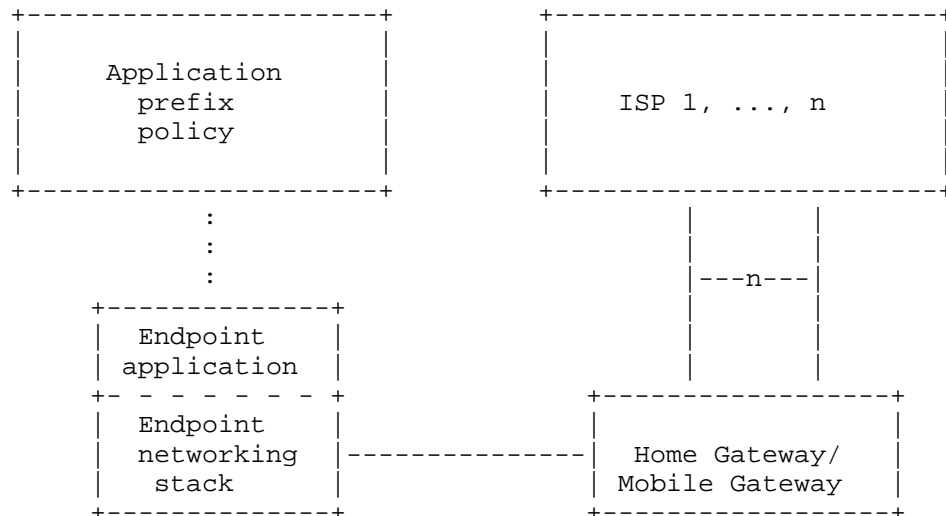


Figure 1

3. Considerations

3.1. Prefix meta-data propagation

The prefix meta-data can be delivered using DHCPv6 prefix delegation and address allocation as elaborated in [I-D.bhandari-dhc-class-based-prefix] or via IPv6 Neighbour discovery (ND) as defined in [I-D.korhonen-6man-prefix-properties].

3.2. Configuring Applications

Applications supporting multiple prefixes obtain the prefixes from the host kernel along with their meta-data.

The policy can then be contained either locally (e.g. If the application is intended only for use within a specific network, linked to a particular ISP comes prepackages with prefix color to

use), or be contained on a remote policy server. The mechanism used to exchange the meta-data information and selection between application/host with a remote server is beyond the scope of this document.

3.3. Application to network stack communication

Once an application has determined the appropriate property and color for its use it has to communicate with the network stack to select the prefix. The host internal data structures need to be extended with the 'prefix property' and the 'prefix color' information associated to the learnt prefix and configured addresses. How this is accomplished is host implementation specific. It is also a host implementation issue how an application can learn or query both properties and color of an address or a prefix. One possibility is to provide such information through the socket API extensions. Other possibilities include the use of e.g., `ioctl()` or NetLink [RFC3549] extensions or by using the IPv6 address scope [RFC4007].

Discussion point: Should prefix property and color be mutually exclusive? This would avoid complexities which takes precedence when one prefix matches color and another matches property. Possibly a prefix may be advertised with both, but the application can only request property or color.

3.4. Default Address Selection

[RFC6724] provides a mechanism for selecting which source address to use, in the absence of an application or upper layer protocol's explicit choice of a legal destination or source address.

The use of prefix meta-data allows an application to express property preferences through socket API extensions, meaning that when used for creating a socket, [RFC6724] source address selection is not required.

If a higher layer protocol or application does not include a prefix property preference when making a create socket request, then source address selection according to [RFC6724] is followed as normal.

3.5. Scope of Prefix Color

Since a home can be connected to multiple ISPs, it is possible that it receives multiple prefixes with the same color from different ISPs. Since the application coloring policy is not received with the color, multiple ISPs may use different coloring policy for a single color. For example: One ISP could use color 50 for video whilst a second ISP is using color 50 for audio.

This section presents some alternatives to handle this problem.

3.5.1. Local scoping

A locally scoped color is a value which is selected by the network and application providers with no central registry. In a multi homed network, this may result in two providers selecting the same color for different behaviors. A color translation could be performed to ensure unique color at the device that connects to multiple providers.

3.5.2. Local scoping with fuzzy matching

To avoid having to maintain multiple colors for each prefix for translating the color, a specific algorithm can be used to determine the new color from the old one on conflict.

For example, when a collision is detected, the new color value may be incremented. Further, colors could be defined to be equally spaced (e.g., 10s or 100s).

Many other encodings are possible as well, as long as obtaining the original color communicated by the ISP may be recovered in the event the application policy server requires this.

3.5.3. Global scoping

A globally scoped color avoids the need for responding to collisions. This can be achieved by disambiguating the color by attaching the domain that provisions the color to the prefix meta-data or by assigning colors from a global registry that comes with the overhead of maintaining such a registry.

3.6. Compatibility with Existing Implementations

The prefix meta-data mechanism that is described in this document provides a way of improving source address selection over the longest-prefix matching method used by [RFC6724].

However, all IPv6 capable hosts deployed at the time of writing do not have the capability of understanding and processing prefix meta-data. This means that any new mechanism must be backwards compatible with existing implementations. Also, clients which understand prefix meta-data need to support applications which do not have meta-data awareness.

There are a number of possible approaches that could be taken here. The following list is included as ideas for further development:

- o In DHCPv6 only clients which request prefixes with meta-data (e.g. signalled through OPTION_ORO in the IA_NA or IA_PD request) will receive them.
- o In case of prefix delegated using IPv6 Neighbour discovery (ND) both forms of prefix i.e with and without meta-data can be offered.
- o If an application makes a socket API request and does not include meta-data as part of the request, follow [RFC6724] source address selection, but remove any prefixes that have meta-data from the list of candidate addresses. It follows that there should be a GU prefix advertised that does not have any meta-data associated that would act as the default choice for non prefix meta-data aware clients and applications.

4. IANA Considerations

Should the global scoping for prefix color be chosen, a new registry should be created by IANA to store colors.

5. Security Considerations

TBD

6. Acknowledgements

The authors would like to acknowledge review and guidance received from

7. Change History (to be removed prior to publication as an RFC)

8. References

8.1. Normative References

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

8.2. Informative References

[I-D.bhandari-dhc-class-based-prefix]
Systems, C., Halwasia, G., Gundavelli, S., Deng, H., Thiebaut, L., and J. Korhonen, "DHCPv6 class based prefix", draft-bhandari-dhc-class-based-prefix-04 (work in progress), February 2013.

[I-D.ietf-dhc-dhcpv4-over-ipv6]

Cui, Y., Wu, P., Wu, J., and T. Lemon, "DHCPv4 over IPv6 Transport", draft-ietf-dhc-dhcpv4-over-ipv6-06 (work in progress), March 2013.

[I-D.jiang-v6ops-semantic-prefix]

Jiang, S., Sun, Q., Farrer, I., and Y. Bo, "A Framework for Semantic IPv6 Prefix", draft-jiang-v6ops-semantic-prefix-03 (work in progress), May 2013.

[I-D.korhonen-6man-prefix-properties]

Korhonen, J., Patil, B., Gundavelli, S., Seite, P., and D. Liu, "IPv6 Prefix Properties", draft-korhonen-6man-prefix-properties-02 (work in progress), July 2013.

[I-D.troan-homenet-sadr]

Troan, O. and L. Colitti, "IPv6 Multihoming with Source Address Dependent Routing (SADR)", draft-troan-homenet-sadr-00 (work in progress), February 2013.

[RFC3549] Salim, J., Khosravi, H., Kleen, A., and A. Kuznetsov, "Linux Netlink as an IP Services Protocol", RFC 3549, July 2003.

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

[RFC3775] Johnson, D., Perkins, C., and J. Arkko, "Mobility Support in IPv6", RFC 3775, June 2004.

[RFC4007] Deering, S., Haberman, B., Jinmei, T., Nordmark, E., and B. Zill, "IPv6 Scoped Address Architecture", RFC 4007, March 2005.

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

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC6724] Thaler, D., Draves, R., Matsumoto, A., and T. Chown, "Default Address Selection for Internet Protocol Version 6 (IPv6)", RFC 6724, September 2012.

Appendix A. Prototype notes

A.1. Homenet prototype implementation notes

This section provides the implementation details of a prototype video application on Android for a Galaxy Nexus device developed for the home network.

A.1.1. Video provider service

A possible use of this prefix coloring is a video service, which requires the network to guarantee a minimal throughput for streaming video. A prefix could be colored by the ISP to indicate that traffic sourced from that prefix will have a certain service level. Using prefix coloring avoids having to set up a separate network for this usage, or implement QoS traffic identification, classification and marking.

An agreement could then be established between the video service provider and the ISP, telling the video provider to use the specific color when streaming video. In the following example, the color 50 was used.

A.1.2. Prefix Color delegation

The CPE routers request prefixes using prefix delegation [RFC3633] with the `OPTION_PREFIX_CLASS` option [I-D.bhandari-dhc-class-based-prefix]. This informs the upstream provider that the CPE supports colored prefixes. If an ISP does not support this option, it will be ignored, and the CPE will only get colorless prefixes. Otherwise, the ISP returns multiple prefixes each with their associated color. A color of '0' is identical to an uncolored prefix, for application compatibility, as explained in Appendix A.1.5. If the CPE does not support colored prefixes, the ISP could decide to delegate a normally colored prefix as an colorless one, though this means hosts will use this prefix according to the default source address selection algorithm, and will not associate any meaning to it.

Once the CPE receives those prefixes, it distributes them, along with their color, using OSPF and the homenet protocols.

[I-D.troan-homenet-sadr] defines "Source Address Dependent Routing" (SADR) which ensures that packets are routed based on their destination as well as source address. SADR is necessary to ensure that a multihomed network using provider aggregatable addresses will send the packet out the right path to avoid violating the provider's ingress filtering. To ensure that those prefixes keep their meaning, Source Address Dependent Routing [I-D.troan-homenet-sadr] is implemented and used.

Colored addresses are advertised to hosts through DHCPv6, to associate the color to the address. Colorless addresses may be distributed through DHCPv6 or through Router Advertisements. Hosts supporting colored prefixes include the `OPTION_PREFIX_CLASS`, and receive colored addresses. For legacy hosts, who do not include this option, there are two possibilities :

- o Those hosts can receive all available prefixes, including colored ones, as uncolored. This allows a legacy host in a fully colored homenet to still have access to IPv6. However, those hosts may use prefixes for the wrong purposes.
- o Those hosts can receive only colorless prefixes. This ensures that a prefix will not be used for the wrong purpose. However, hosts in a fully colored environment will not get access to IPv6. This can however be what the ISP originally intended, for example if the ISP does not provide access to the IPv6 Internet, but uses IPv6 for wall gardened services, which their specific devices know how to use.

A.1.3. Configuring Applications

Applications supporting multiple prefixes obtain the prefixes from the host kernel, along with their color.

The policy can be contained either in a local database (e.g. If the application is intended only for use within a specific network, linked to a particular ISP), or be contained on a distant server.

For applications that do not contain a local database, an HTTP POST request is sent to a predefined server using a colorless prefix. This server, through means that are out of the scope of this document, selects the most appropriate color for the URIs used by the application. It then returns an XML document containing the mapping between the URIs and the colors. URIs in this document MAY use wild cards.

When the application is started, it sends the available prefixes and their color to the video provider server which answers with a wild card URI videos.example.com associated to color 50. In this example application it receives:

```
<?xml version="1.0" encoding="UTF-8"?>
<mappings>
  <mapping>
    <URI>*://audio.example.com/*</URI>
    <color>40</color>
  </mapping>
  <mapping>
    <URI>rtsp://video.example.com/*</URI>
    <color>50</color>
  </mapping>
</mappings>
```

The server is expected to know the application, and thus to list all URIs that could be of use to the application. The application will not ask the server if it has to contact an address not in the list and will use the colorless prefix. This avoids an additional delay when trying to contact an unlisted URI.

Example: While the application is browsing the video list, it is using www.example.com, and thus the colorless prefix. However as soon as a video is chosen, it starts streaming from videos.example.com, and asks to connect to host videos.example.com with color 50, indicating that it wishes to use the colored prefix.

A.1.4. Android DHCPv6

Considering that this prototype is being implemented on Android, the first step is to get a running DHCPv6 client on Android, with support for the OPTION_PREFIX_CLASS option.

The odhcp6c client, which already supports OPTION_PREFIX_CLASS, has been ported to Android, and is set to run in parallel to the dhcpcd client used for DHCP. The success of any of the two clients results in the success of the WiFi connection, so as to support IPv6 only networks.

This client configures the IPv6 addresses using calls to IP address, which is modified to support the addition of a class option to set the prefix color.

A.1.5. Application to network stack communication

Once an application has received the appropriate color for its use, in this prototype it specifies the prefix it wishes to use by using the IPv6 address scope [RFC4007]. When resolving this address, the standard library then adds this information in the address information it returns, using the scope field, allowing the kernel to appropriately select the source IP when connecting. For this reason, a color of 0 is identical to an colorless prefix.

In the example, when downloading from video.example.com, the application would request a connection to video.example.com%50.

This allows the user to override the application's default simply by specifying a color in the scope of the URI it is trying to access, and requires little to no change in applications to support it. Applications that allow scope ids do not need to be modified in order to allow the user to use multiple prefixes (though it is then up to the user to select its color). A web browser that allows scope id would allow the user to add a color to the URI, without requiring any modifications.

A.1.6. Android kernel

To reduce the amount of modifications needed by the applications to support this prefix coloring, we need to avoid having to bind to the address in the colored prefix before initiating the connection. The kernel is expected to choose the correct source address when a colored destination is used.

This implies storing the color in the kernel, along with the address, which is done using a new attribute IFA_color to the netlink message RTM_NEWADDR, used by ip address. Setting a colored prefix using ioctl is not supported.

Since colors are put in the scope id part of the destination address, we continue to use the scope element of the sockaddr_in6 structure to store the color when sending connect messages to the kernel. The scope is only used when considering local addresses, so we interpret the presence of a scope on a non link-local address to be a color. Colors can not be assigned to link-local addresses, but since they are on the same link, source address shouldn't impact how the network treats packets. When selecting the source address, we then discard all addresses which do not have the correct color.

A.1.7. Limitations of the current prototype

It does not implement any duplicate color detection. Colors are considered to be unique within the home, and to correspond to the original color provided by the ISP. This is compatible with Global scoping. No changes would be required to the host in order to support Local scoping with fuzzy matching, but OSPF would need to detect collisions, and the server would need to recalculate the original color before making a decision. In this implementation, hosts that do not support colors do not receive colored prefixes.

Authors' Addresses

Maico Le Pape
Cisco Systems
Paris
FR

Email: maico@maicolepape.org

Shwetha Bhandari
Cisco Systems
Cessna Business Park, Sarjapura Marathalli Outer Ring Road
Bangalore, KARNATAKA 560 087
India

Email: shwethab@cisco.com

Ian Farrer
Deutsche Telekom AG
GTN-FM4, Landgrabenweg 151
Bonn 53227
Germany

Email: ian.farrer@telekom.de

HOMENET
Internet-Draft
Intended status: Standards Track
Expires: January 06, 2014

D. Migault (Ed)
Francetelecom - Orange
W. Cloetens
SoftAtHome
C. Griffiths
Dyn
R. Weber
Nominum
July 05, 2013

IPv6 Home Network Naming Delegation
draft-mglt-homenet-front-end-naming-delegation-02.txt

Abstract

CPEs are designed to provide IP connectivity to home networks. Most CPEs assigns 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. More specifically, CPE have been designed neither to host a service exposed on the Internet, nor to support heavy operations like zone signing. Both 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 outsource both DNSSEC management and naming service 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 06, 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	3
3. Terminology	4
4. Architecture Overview	5
5. Architecture Description	7
5.1. CPE and Public Authoritative Servers Synchronization . .	7
5.1.1. Synchronization with a Hidden Master	7
5.1.2. Securing Synchronization	8
5.2. DNS Homenet Zone configuration	9
5.3. DNSSEC outsourcing configuration	10
5.4. CPE Security Policies	11
6. Homenet Naming Configuration	11
7. Security Considerations	12
7.1. Names are less secure than IP addresses	13
7.2. Names are less volatile than IP addresses	13
8. IANA Considerations	13
9. Acknowledgment	13
10. Normative References	14
Appendix A. Document Change Log	14
Authors' Addresses	16

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 from the Internet and into the Home Network. End Users to access services hosted in the Home Network with IPv6 addresses would 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 other words, the CPE is the natural candidate for setting the DNS(SEC) zone file.

CPEs are usually low powered devices designed for the Home Network, but not for heavy traffic. CPEs can host the Naming Service for the Home Network but should not be exposed on the Internet. This would expose the CPE to resource exhaustion. As a consequence, it may isolate the Home Network from the Internet and affects the services hosted by the CPEs, thus affecting Home Network communications. As a result, CPE SHOULD NOT host the Naming Service of the Home Network for resolutions coming from the Internet.

Similarly, CPEs have not been designed to handle heavy computation such as DNSSEC zone signing. Such operations could also result in CPE resource exhaustion. As a consequence, resource expensive operations such as zone signing SHOULD NOT be handled by the CPE, but SHOULD be handled by other third party.

In addition to heavy operations such as zone signing, DNSSEC comes with complex configurations as well as complex operation management like (DNSSEC secure delegation, DNSSEC key roll over, DNSSEC zone updates). These operations can hardly be understood by the average end user, and a misconfiguration MAY result in invalid naming resolutions that MAY make an host, or the whole home network unreachable. As a consequence, DNSSEC management operations SHOULD NOT be handled by the average end user, but SHOULD be handled by a third party.

The goal of this document is to describe an architecture where the CPE outsources the authoritative naming service and DNSSEC zone management to a third party designated as Public Authoritative Servers. This document describes the involved protocols as well as their respective configurations to properly set the homenet naming architecture.

The document is organized as follows. Section 4 provides an overview of the homenet naming architecture and presents the CPE and the Public Authoritative Server that handles the authoritative naming service of the home network as well as DNSSEC management operations on behalf of the CPE. Section 5 describes in details protocols and configurations to set the homenet naming architecture. Section 6 sums up the various configuration parameters that MAY be filled by the end user on the CPE for example via a GUI. Finally Section 7 provides security considerations.

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.
- 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.

4. Architecture Overview

Figure 1 provides an overview of the homenet naming architecture.

The CPE is in charge of building the DNS Homenet Zone that contains all FQDN bindings of the home network. The home network is associated to a FQDN, the Registered Homenet Domain (example.com). Any node in the home network is associated to a FQDN (node1.example.com) that MAY be provided via DHCP or statically configured on the CPE via a GUI for example.

The goal of the homenet naming architecture is that the CPE does not handle any DNSSEC operations and does not host the authoritative naming service while FQDNs in the Homenet Zone can be resolved with DNSSEC by any node on the Internet.

In order to achieve this goal, when a node on the Internet sends a DNS(SEC) query like for node1.example.com, this DNS(SEC) query MUST be treated by a third party designated in figure 1 as the Public Authoritative Servers.

The Public Authoritative Servers are in charge of DNS(SEC) traffic for the Registered Homenet Domain (example.com) as well as all DNSSEC management operations like zone signing, key rollover. The DNSSEC zone hosted by the Public Authoritative Servers is called the DNSSEC Public Zone.

The purpose of our architecture is to describe how the CPE can outsource the DNS Homenet Zone hosted on the CPE to the DNSSEC Public Zone hosted on the Public Authoritative Servers. This includes description of the synchronization protocols between the CPE and the Public Authoritative Servers in Section 5.1 as well as configurations of the DNS Homenet Zone Section 5.2.

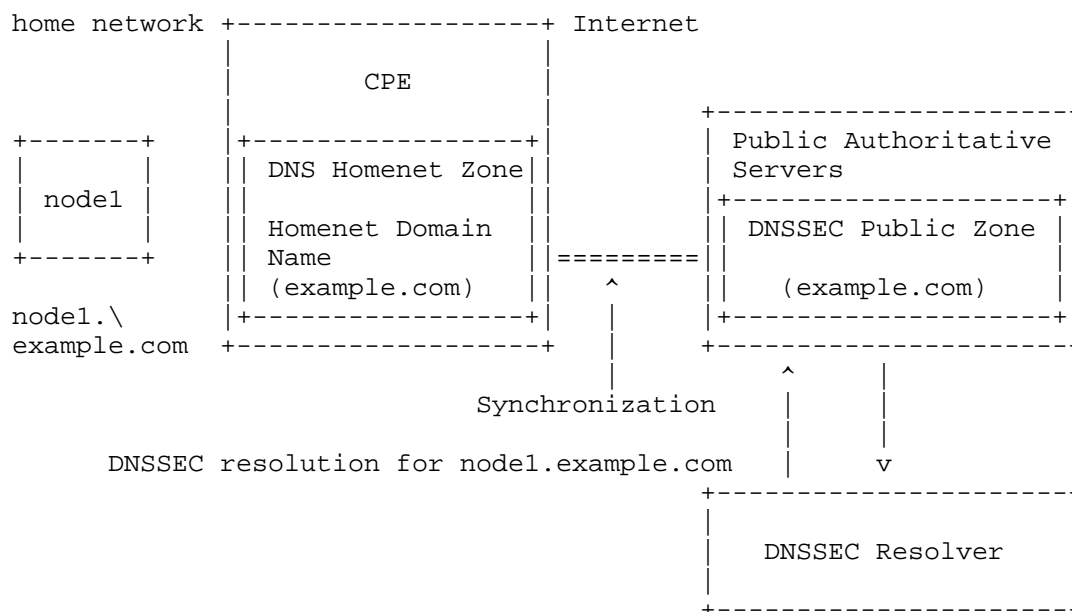


Figure 1: Homenet Naming Architecture Description

The content of the DNS Homenet Zone is out of the scope of this document. The CPE MAY host 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 does not address this issue.

CPE MAY chose to host an authoritative naming server for the home network. Whether this service is implemented or not on the CPE is out of the scope of this document. Some implementations MAY chose to set a DNS authoritative server for the DNS Homenet Zone for resolutions coming from the home network. Other implementations MAY chose to synchronize the DNSSEC zone on the Public Authoritative Servers to provide DNSSEC responses. This latest option MAY require specific configurations on the Public Authoritative Servers.

Similarly, CPE MAY host a DNS(SEC) resolution service for nodes in the home network. There are multiple ways to configure the resolver service on the CPE. Detailing these various configurations is out of the scope of this document, and is considered as an implementation issue. Some implementers MAY chose to forward DNS(SEC) queries from the home network to the resolving server of its ISP or any other public resolver. In that case, the DNS(SEC) response from the Public

Authoritative Servers is forwarded to the home network, which provide DNS and DNSSEC resolution for the home network. Alternative implementations MAY chose to lookup in the DNS Homenet Zone, and thus provide only DNS responses in the home network. Other implementation MAY chose to synchronize the DNSSEC Public Zone on the CPE either using DNS master slave mechanisms, or by caching the whole zone. This latest option MAY require some additional configuration the Public Authoritative Servers.

5. Architecture Description

This section describes how the CPE and the Public Authoritative Servers SHOULD be configured to outsource authoritative naming service as well as DNSSEC management operations. Section 5.1 describes how a secure synchronization between the CPE and the Public Authoritative server is set. Section 5.2 provides guide lines for the DNS Homenet Zone set in the CPE and uploaded on the Public Authoritative Servers. Section 5.3 describes DNSSEC settings on the Public Authoritative Servers. Finally, Section 5.4 provides the security policies that SHOULD be set on the CPE.

5.1. CPE and Public Authoritative Servers Synchronization

5.1.1. Synchronization with a Hidden Master

Uploading and dynamically updating the zone file on the Public Servers can be seen as zone provisioning between the CPE (Hidden Master) and the Public Server (Slave Server). This can be handled either in band or out of band. DNS dynamic update [RFC2136] may be used. However, in this section we detail how to take advantage of the DNS slave / master architecture to deploy updates to public zones.

The Public Authoritative Server 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 only expected to initiate AXFR [RFC1034], IXFR [RFC1995] transfers to configured slave DNS servers. The Hidden Master Server SHOULD send NOTIFY messages [RFC1996] in order to update Public DNS server zones as updates occur.

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 Servers, 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 Servers. 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.

5.1.2. 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 DNS(SEC) Servers.

TSIG [RFC2845] can be used to secure the DNS communications between the CPE and the Public DNS(SEC) Servers. TKEY [RFC2931] can be used for re-keying the key used for TSIG. The advantage of this mechanism is that this mechanisms are only associated with the DNS application. Not relying on shared libraries ease testing and integration. On the other hand, using TSIG and TKEY requires that this mechanism is 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.

5.2. DNS Homenet Zone configuration

As depicted in figure 1, the DNSSEC Public Zone is hosted on the Public Authoritative Server, whereas the DNS Homenet Zone is hosted on the CPE. As a result, the CPE MUST configure the DNS Homenet Zone as if the DNS Homenet Zone were hosted by the Public Authoritative Servers instead of the CPE.

If one considers the case where the CPE has a single Homenet Domain Name and has an agreement with a single Public Authoritative Server. In that case, the DNS Homenet Zone SHOULD configure its Name Server RRset and Start of Authority with the ones associated to the Public Authoritative Servers. This is illustrated in figure 2. public.autho.servers.example.net is the domain name associated to the Public Authoritative Server, and IP1, IP2, IP3, IP4 are the IP addresses associated.

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

@ IN SOA public.autho.servers.example.net
    user.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.autho.servers.example.net A @IP1
public.autho.servers.example.net A @IP2
public.autho.servers.example.net AAAA @IP3
public.autho.servers.example.net AAAA @IP4
```

Figure 2: DNS Homenet Zone

When the end user considers multiple Public Authoritative Servers for a given Registered Homenet Domain, the DNS Homenet Zone MAY contain all associated Name Servers and IP addresses.

Some additional verification can check whether the CPE IP address is mentioned in the Public Zone file, and raise a warning to the End User.

5.3. DNSSEC outsourcing configuration

In this document we assumed that the Public Authoritative Server signs the DNS Homenet Zone. Multiple variants MAY be proposed by the Public Authoritative Servers. Public Authoritative Servers MAY propose to sign the DNS Homenet Zone with keys generated by the Public Authoritative Servers and unknown to the CPE. Alternatively some MAY propose the end user to provide the private keys. Although not considered in this document some end user MAY still prefer to sign their zone with their own keys they do not communicate to the Public Authoritative Servers. All these alternatives result from a negotiation between the end user and the Public Authoritative Servers. This negotiation is performed out-of-band and is out of scope of this document.

In this document, we consider that the Public Authoritative Server has all the necessary cryptographic elements to perform zone signing and key management operations.

Note that Public Authoritative Servers described in this document accomplish different functions, and thus different entities MAY be involved.

- DNS Slave function synchronizes the DNS Homenet Zone between the CPE and the Public Authoritative Servers. The DNS Homenet Zone on the Public Authoritative Servers is not available, and the Public Authoritative Server MUST NOT address any DNS queries for that zone. As a result, the Public Authoritative Servers MAY chose a dedicated set of servers to serve the DNS Homenet Zone: the Public Authoritative Name Server Set.
- DNS Zone Signing function signs the DNS Zone Homenet Zone to generate an DNSSEC Public Zone.
- DNSSEC Authoritative Server hosts the naming service for the DNSSEC Public Zone. Any DNS(SEC) query associated to the Homenet Zone SHOULD be done using the specific servers designated as the Public Authoritative Master(s).

5.4. 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. Homenet Naming Configuration

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 Servers MAY be defined with the following parameters. These parameters are necessary to establish a secure channel between the CPE and the Public Authoritative Server, and to set the appropriated DNS Homenet Zone file:

- 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 corresponds 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 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.

7. Security Considerations

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

7.1. 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.

7.2. 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.

8. IANA Considerations

This document has no actions for IANA.

9. 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.

10. Normative References

- [RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, 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.
- [RFC2845] Vixie, P., Gudmundsson, O., Eastlake, D., and B. Wellington, "Secret Key Transaction Authentication for DNS (TSIG)", RFC 2845, May 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.
- [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.

Appendix A. Document Change Log

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

-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
Francetelecom - 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

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: December 28, 2013

M. Stenberg
June 26, 2013

Hybrid Unicast/Multicast DNS-Based Service Discovery Auto-Configuration
Using OSPFv3
draft-stenberg-homenet-dnssdext-hybrid-proxy-ospf-00

Abstract

This document describes how a proxy functioning between Unicast DNS-Based Service Discovery and Multicast DNS can be automatically configured using automatically configured routing protocol or some other network-level state sharing mechanism. Zero-configuration OSPFv3 is used to describe one concrete way to implement this scheme.

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 28, 2013.

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. Introduction	3
2. Requirements language	3
3. Hybrid proxy - what to configure	3
3.1. Conflict resolution with OSPFv3	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. Router name (scheme 2)	5
3.3.3. Link name (scheme 2)	5
4. OSPFv3 auto-configuration TLVs	6
4.1. DNS Delegated Zone TLV	6
4.2. Domain Name TLV	7
4.3. Router Name TLV	8
4.4. DNS Server TLV	8
5. Desirable router behavior	9
5.1. DNS search path	9
5.2. Hybrid proxy	9
5.3. OSPFv3 daemon	10
6. Security Considerations	10
7. IANA Considerations	10
8. References	11
8.1. Normative references	11
8.2. Informative references	11
Appendix A. Example configuration	12
A.1. Topology	12
A.2. OSPFv3-DNS interaction	12
A.3. OSPFv3 state	13
A.4. DNS zone	14
A.5. Interaction with hosts	14
Appendix B. Implementation	15
Appendix C. Why not just proxy Multicast DNS?	15
C.1. General problems	15
C.2. Stateless proxying problems	16
C.3. Stateful proxying problems	16
Appendix D. Acknowledgements	17
Author's Address	17

1. Introduction

Section 3 ("Hybrid Proxy Operation") of [I-D.cheshire-mdnsexthybrid] 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 DNS servers require, as well as how it can be provided using auto-configured OSPFv3 described in [I-D.ietf-ospf-ospfv3-autoconfig] and a naming scheme which does not even need to be same across the whole covered network. The scheme can be used to provision both forward and reverse DNS zones which employ hybrid proxy for heavy lifting.

While this document describes the data to be transferred in auto-configured OSPFv3 TLVs, in principle same scheme could work across other routing protocols, or even some non-routing protocol, as long as the consistent state for it is available across the whole covered network (by for example site-scoped multicast, or some other flooding scheme).

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 in OSPFv3 is described in Section 4. Finally, some overall notes on desired behavior of different router 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-mdnsexthybrid] describes just that there have to be NS records pointing to hybrid proxy responsible for each link within the covered network.

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 OSPFv3 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 router which is connected to a link, and running hybrid proxy. We also want to populate reverse DNS zone similarly per each prefix in use. Links' forward DNS zone names should be unique.

There should be DNS-SD domain search list provided for the network's domain which contains domain for each physical link only once, regardless of how many routers 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 domain lists. Typically, it contains only that the 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 with OSPFv3

Any naming-related choice on a router may have conflicts in the network.

We use similar conflict resolution scheme as described in the prefix assignment draft[I-D.arkko-homenet-prefix-assignment]. That is, if a conflict is encountered, the router with highest router ID MUST keep the name they have chosen. The one(s) with lower router ID 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 router 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. Router and link name - (link).(router).(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 router 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").

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

3.3.2. Router name (scheme 2)

Recommendation is to use some short form which indicates the type of router 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 practise, though, providing some other records indicating exact router information (and access to management UI) might be more sensible.

If scheme 2 is used, and there is no desire to implement conflict resolution related TLV described in Section 4.3, a safe default might be to default to router ID; that is, use as router name value such as `r-(router ID as single 32-bit number)`. It is guaranteed to be unique across the network, but not as user-friendly as the descriptive router name promoted here.

3.3.3. Link name (scheme 2)

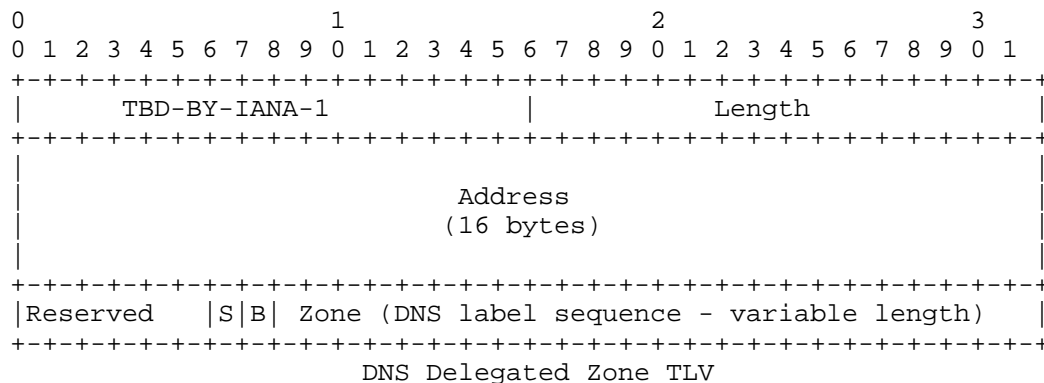
Recommendation for (link) portion of (link).(router).(domain) is to use either physical network layer type as base, possibly even just interface name on the router, if it's descriptive enough, for example, eth0.router1.example.com and wlan0.router1.example.com may be good enough.

4. OSPFv3 auto-configuration TLVs

To implement this specification fully, support for following three different new OSPFv3 auto-configuration 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 routers 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 router with highest router ID on the link should publish this TLV.



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 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 indicates that this delegated zone should be included in network's DNS-SD list of domains recommended for browsing 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.

4.2. Domain Name TLV

This TLV is used to indicate the base (domain) to be used for the network. If multiple routers advertise different ones, the conflict resolution rules in Section 3.1 should result in only the one with highest router ID 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 routers.

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 routers without manual configuration options, being able to read the domain name provided by a different router could make the user experience better due to consistent naming of zones across the network.

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	2	3	4	5	6	7	8	9
TBD-BY-IANA-2										Length																													
Domain (DNS label sequence - variable length)																																							

Domain Name TLV

Like the Zone field in Section 4.1, the Domain Name TLV's contents are encoded as label sequence.

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

4.3. Router Name TLV

This TLV is used to advertise a router's name. After the conflict resolution procedure described in Section 3.1 finishes, there should be exactly zero to one routers publishing each router name.

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

```

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
+-----+-----+-----+-----+-----+-----+-----+-----+
|          TBD-BY-IANA-3          |          Length          |
+-----+-----+-----+-----+-----+-----+-----+-----+
|Name (not even null terminated - variable length)          |
+-----+-----+-----+-----+-----+-----+-----+-----+
                                Router Name TLV

```

If the router 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 routers.

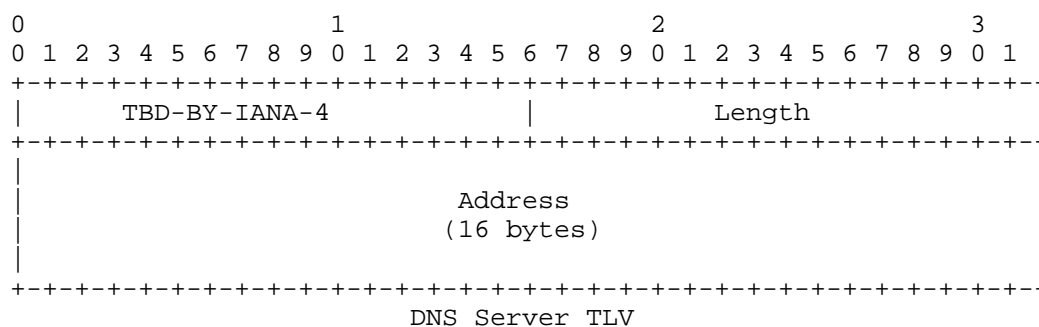
4.4. DNS Server TLV

This TLV is used to announce address of a fallback recursive DNS server (provided by e.g. ISP). If the DNS server implementations used in the network are not full recursive resolver implementations, they may respond to network-specific queries within network, and forward the rest to the provided DNS servers. Even recursive resolver implementations may want to use these servers, if available, for better caching and therefore more responsive user experience.

Typically, these addresses are gleaned from (for example) a DHCPv4/DHCPv6 exchange, or from Router Advertisements.

Any router on the home network can publish 0-N of these TLVs, and the order in which they are used is not defined (we assume that the DNS view of the world is consistent; this may not be true in all cases).

This TLV SHOULD be supported by routers, but the routers (and DNS servers in the network) MUST be able to cope even in the absence of the TLV. This can be handled by (for example) DNS servers providing recursive resolving fallback functionality, or defaulting to some known global recursive resolver.



The address may be again either IPv4 or IPv6 address, with the IPv4 address encoded under the `::FFFF:/96` prefix.

It is important to note that if the network's domain forward or reverse resolution will not work globally, using network-external DNS server directly is not good. Therefore the network's local DNS servers should be announced to hosts in e.g. DHCPv4/DHCPv6/RA, and then only those DNS servers can use the contents of this TLV as fallback for non-local resolution if so desired. How these local DNS server addresses are propagated within home network is outside the scope of this document

5. Desirable router behavior

5.1. DNS search path

The routers 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 OSPFv3 daemon. 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 SHOULD 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. OSPFv3 daemon

OSPFv3 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 that it does not have highest router ID that supports this specification. (Support for this specification can be deduced by e.g. presence of any TLVs from this draft advertised by a router.)

OSPFv3 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 routing 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 routing protocol set-up which has authentication.

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

7. IANA Considerations

This document makes two allocations out of the OSPFv3 Auto-Configuration (AC) LSA TLV namespace
[I-D.ietf-ospf-ospfv3-autoconfig]:

- o The DNS Delegated Zone TLV in Section 4.1 takes the value TBD-BY-IANA-1 (suggested value is 4).
- o The Domain Name TLV in Section 4.2 takes the value TBD-BY-IANA-2 (suggested value is 5).
- o The Router Name TLV in Section 4.3 takes the value TBD-BY-IANA-3 (suggested value is 6).
- o The DNS Server TLV in Section 4.4 takes the value TBD-BY-IANA-4 (suggested value is 7).

8. References

8.1. Normative references

- [I-D.cheshire-mdnsexthybrid]
Cheshire, S., "Hybrid Unicast/Multicast DNS-Based Service Discovery", draft-cheshire-mdnsexthybrid-01 (work in progress), January 2013.
- [I-D.ietf-ospf-ospfv3-autoconfig]
Lindem, A. and J. Arkko, "OSPFv3 Auto-Configuration", draft-ietf-ospf-ospfv3-autoconfig-02 (work in progress), April 2013.
- [RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, February 2013.
- [RFC6763] Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", RFC 6763, February 2013.

8.2. Informative references

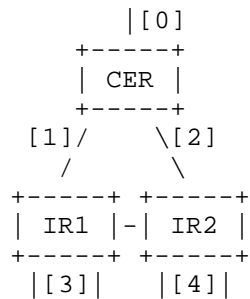
- [I-D.arkko-homenet-prefix-assignment]
Arkko, J. and A. Lindem, "Prefix Assignment in a Home Network", draft-arkko-homenet-prefix-assignment-01 (work in progress), October 2011.
- [RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, December 2003.

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

Appendix A. Example configuration

A.1. Topology

Let's assume home network that looks like this:



We're not really interested about links [0], [1] and [2], or the links between IRs. Given the optimization described in Section 4.1, they should not produce anything to OSPF 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::/48 prefix to be delegated in the home via [0].

A.2. OSPFv3-DNS interaction

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

1. OSPFv3 auto-configuration takes place, routers get their router IDs. 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:db8:1:1::/64 for LAN and 2001:db8:1:2::/64 for WLAN. IR2 winds up with 2001:db8:2:1::/64 for LAN and 2001:db8:2:2::/64 for WLAN.
3. IR1 is assumed to be reachable at 2001:db8:1:1::1 and IR2 at 2001:db8:2:1::1.

4. Each router wants to be called 'router' due to lack of branding in drafts. They announce that using the router 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 highest router ID, it becomes "router". CER and IR2 have to rename, and (depending on timing) one of them becomes "router-2" and other one "router-3". Let us assume IR2 is "router-2". During conflict resolution, each router 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. OSPFv3 state

Once there is no longer any conflict in the system, we wind up with following TLVs within OSPFv3 (RN is used as abbreviation for Router Name, and DZ for Delegated Zone TLVs):

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

(from IR1)
RN {name="router"}

DZ {address=2001:db8:1:1::1, b=1,
    zone="lan.router.example.com."}
DZ {address=2001:db8:1:1::1,
    zone="1.0.0.0.1.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa."}

DZ {address=2001:db8:1:1::1, b=1,
    zone="wlan.router.example.com."}
DZ {address=2001:db8:1:1::1,
    zone="2.0.0.0.1.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa."}

(from IR2)
RN {name="router-2"}

DZ {address=2001:db8:2:1::1, b=1,
    zone="lan.router-2.example.com."}
DZ {address=2001:db8:2:1::1,
    zone="1.0.0.0.2.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa."}

DZ {address=2001:db8:2:1::1, b=1,
    zone="wlan.router-2.example.com."}
DZ {address=2001:db8:2:1::1,
```



```
zone="2.0.0.0.2.0.0.0.8.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 routers, 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.router  
b._dns_sd._udp PTR wlan.router
```

```
b._dns_sd._udp PTR lan.router-2  
b._dns_sd._udp PTR wlan.router-2
```

```
router AAAA 2001:db8:1:1::1  
router-2 AAAA 2001:db8:2:1::1
```

```
router NS router  
router-2 NS router-2
```

```
1.0.0.0.1.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa. NS router.example.com.  
2.0.0.0.1.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa. NS router.example.com.  
1.0.0.0.2.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa. NS router-2.example.com.  
2.0.0.0.2.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa. NS router-2.example.com.
```

Internally, the router 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 routers? 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 (and transitively also of [I-D.cheshire-mdnsexthybrid]) at `hnet-core` github repository [1] which contains variety of other homenet WG-related things' implementation too.

`hp.lua` binary can be used to start hybrid proxy either as one-router stand-alone implementation (that can be used to e.g. use statically configured DNS zones), or as part of zeroconf OSPFv3 configured set of proxies.

Sample usage case:

```
# sudo lua hp.lua eth0 eth1
.. no output ..
```

Given the command, hybrid proxy is started for interfaces `eth0` and `eth1`, and it will publish DNS zones `l-eth0.r-router.home`, `l-eth1.r-router.home` (and home zone with relevant DNS-SD sub-zone, and default forward behavior) in DNS port. It has `-h` option for seeing various options that can be set, notable one being `--ospf` (use OSPFv3 autoconfigured hnet infrastructure).

Disclaimer: The set-up of third-party libraries etc to get the implementation running may be painful and is omitted here. Use of ready UML NetKit-based test environment or building image for a real router using hnet github repository [2] is recommended.

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 ordering which 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 workaround for this is to make sure that in conflict resolution, propagated resources always loses. Due to conflict handling ordering logic, and the arbitrary order in which the original records may be in, this is non-trivial.

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 and Shwetha Bhandari for review comments.

Author's Address

Markus Stenberg
Helsinki 00930
Finland

Email: markus.stenberg@iki.fi