

Network Working Group
Internet-Draft
Intended status: Informational
Expires: August 18, 2014

T. Chown, Ed.
University of Southampton
J. Arkko
Ericsson
A. Brandt
Sigma Designs
O. Troan
Cisco Systems, Inc.
J. Weil
Time Warner Cable
February 14, 2014

IPv6 Home Networking Architecture Principles
draft-ietf-homenet-arch-12

Abstract

This text describes evolving networking technology within residential home networks with increasing numbers of devices and a trend towards increased internal routing. 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 August 18, 2014.

Copyright Notice

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

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

Table of Contents

1. Introduction	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 Principles	11
3.1. General Principles	12
3.1.1. Reuse existing protocols	12
3.1.2. Minimise changes to hosts and routers	12
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	22
3.3.4. Homenet realms and borders	22
3.3.5. Configuration information from the ISP	23
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	31
3.6.2. Filtering at borders	31
3.6.3. Partial Effectiveness of NAT and Firewalls	32
3.6.4. Exfiltration concerns	32
3.6.5. Device capabilities	33
3.6.6. ULAs as a hint of connection origin	33
3.7. Naming and Service Discovery	33
3.7.1. Discovering services	33
3.7.2. Assigning names to devices	34
3.7.3. The homenet name service	35
3.7.4. Name spaces	36
3.7.5. Independent operation	38
3.7.6. Considerations for LLNs	38
3.7.7. DNS resolver discovery	39
3.7.8. Devices roaming to/from the homenet	39
3.8. Other Considerations	39
3.8.1. Quality of Service	39
3.8.2. Operations and Management	40
3.9. Implementing the Architecture on IPv6	41
4. Conclusions	41
5. Security Considerations	42
6. IANA Considerations	42
7. References	42
7.1. Normative References	42
7.2. Informative References	42
Appendix A. Acknowledgments	45
Appendix B. Changes	45
B.1. Version 12	46
B.2. Version 11 (after IESG review)	46
B.3. Version 10 (after AD review)	46
B.4. Version 09 (after WGLC)	46
B.5. Version 08	47
B.6. Version 07	47
B.7. Version 06	48
B.8. Version 05	48
B.9. Version 04	49
B.10. Version 03	49
B.11. Version 02	50
Authors' Addresses	51

1. Introduction

This document focuses on evolving networking technology within residential home networks with increasing numbers of devices and a trend towards increased internal routing, 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 should be avoided, 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 aiming at a more consistent solution that addresses as many of the identified requirements as possible. 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. Likewise, we also do not specify the whole design of a homenet router from top to bottom, rather we focus on the Layer 3 aspects. This means that Layer 2 is largely out of scope, we're assuming a data link layer that supports IPv6 is present, and that we react accordingly. Any IPv6-over-Foo definitions occur elsewhere.

[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 [RFC7084], 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 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.
- o Guest network: A part of the home network intended for use by visitors or guests to the home(net). Devices on the guest network may typically not see or be able to use all services in the home(net).
- 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 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 [RFC6887].
- 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 Address [RFC4193].
- o VM: Virtual machine.

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 home automation or corporate extension LANs (whereby a home worker can have their corporate network extended into the home using a virtual private network, commonly presented as one port on an Ethernet device) 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 makes it possible 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 unlimited, 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 and possibly malicious 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.

This topic is discussed further in Section 3.6.1.

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 home automation networks are being set up in a new home/deployment (as early as during construction of the home), such networks will likely need to use their own /48 ULA prefix. Depending upon circumstances beyond the control of the owner of the homenet, it may be impossible to renumber the ULA used by the home automation network 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 an IPv6 equivalent of a traditional IPv4 NAT [RFC3022], or of NPTv6 prefix-based NAT [RFC6296]. When an IPv6 node in a homenet has both a ULA and a globally unique IPv6 address, it should only use its ULA address internally, and use its additional globally unique IPv6 address as a source address for external communications. This should be the natural behaviour given support

for Default Address Selection for IPv6 [RFC6724]. 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. Further, the homenet border router(s) should filter packets with ULA source/destination addresses as discussed in Section 3.4.2.

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.

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.

2.6. IPv6-only operation

It is likely that IPv6-only networking will be deployed first in new home network deployments, often referred to as 'greenfield' scenarios, where there is no existing IPv4 capability, 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.

Some specific transition technologies which may be deployed by the homenet's ISP are discussed in [RFC1918]. In addition, certain other functions may be desirable on the CER, e.g., to access content in the IPv4 Internet, NAT64 [RFC6144] and DNS64 [RFC6145] may be applicable.

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 Principles

The aim of this text is to outline how to construct advanced IPv6-based home networks involving multiple routers and subnets using standard IPv6 addressing and protocols [RFC2460] [RFC4291] as the basis. As described in Section 3.1, solutions should as far as possible re-use existing protocols, and minimise changes to hosts and routers, but some new protocols, or extensions, are likely to be required. 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 protocol 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

In order to maximise deployability of new homenets, 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. There may be cases where changes are unavoidable, e.g., to allow a given homenet routing protocol to be self-configuring.

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., deactivating 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

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

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 Architectures described in [RFC6204]

and its successor [RFC7084] 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 may need to manage connection-oriented state mappings on the same CER as used for subsequent traffic flows.

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).

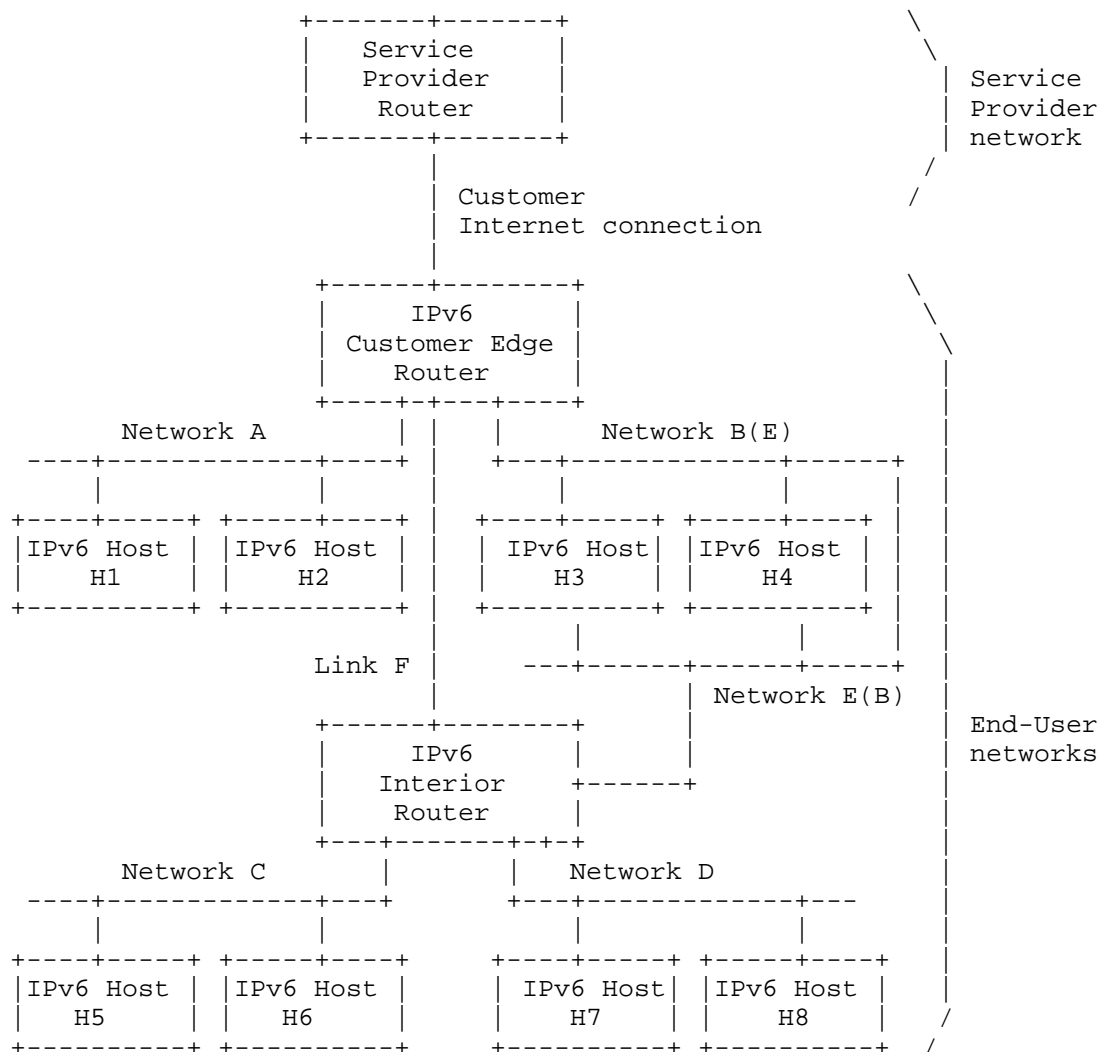


Figure 1

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

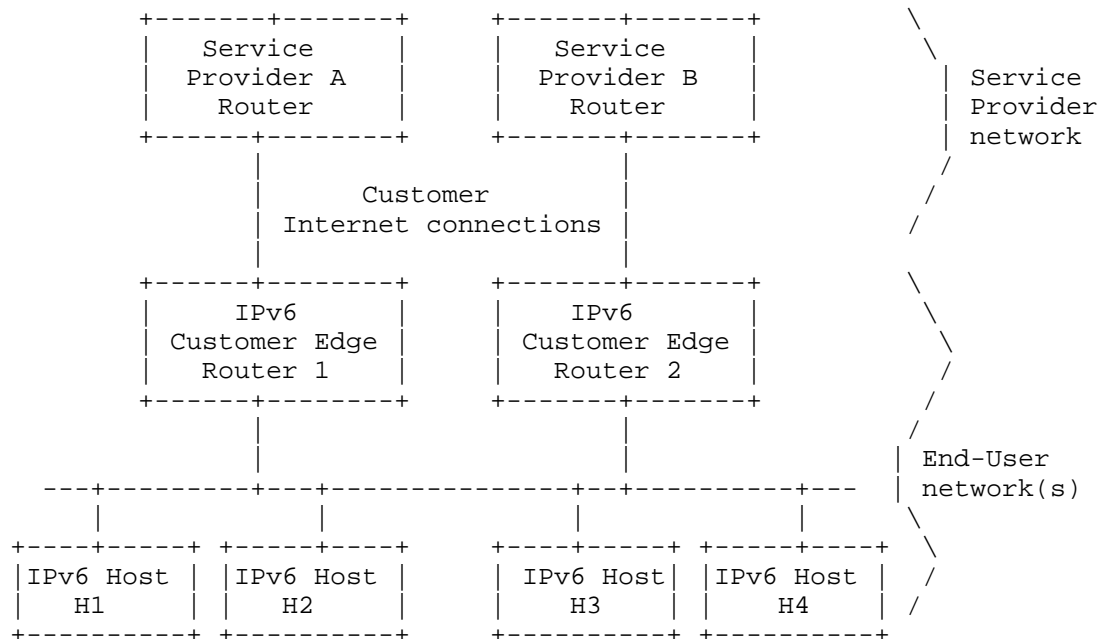


Figure 2

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 prefixes, 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

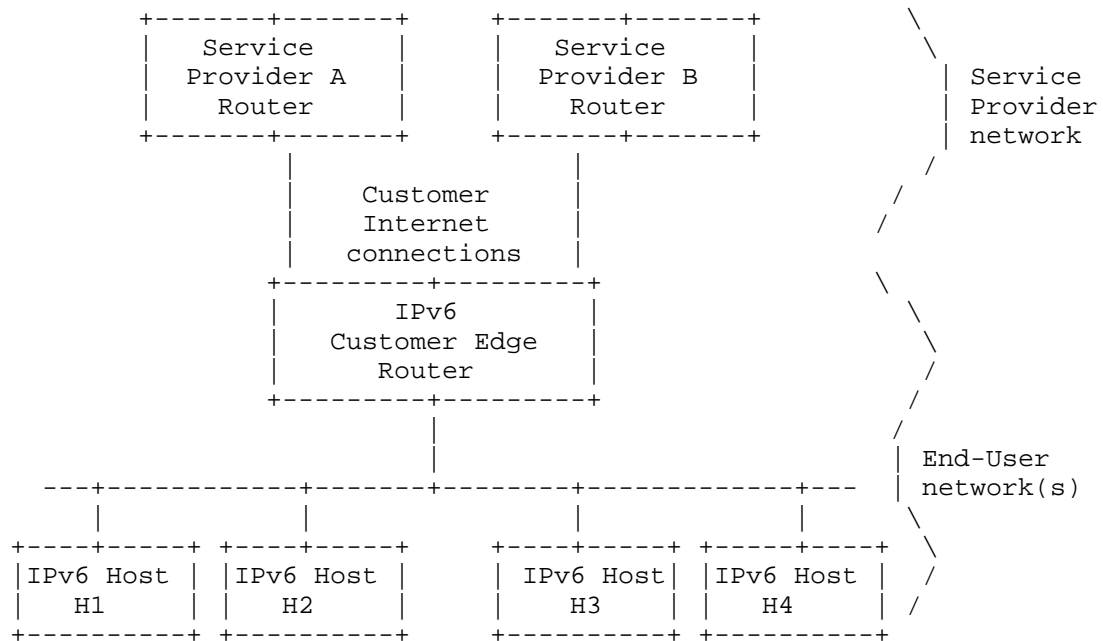


Figure 3

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.

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-based 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 ingress 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 throughout the homenet for routing which is based on the source as well as the destination address of each packet. 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. As per [RFC3002] (section 4.2.1) we do not specifically target walled garden multihoming as a goal of this document.

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 infrastructure 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 homenet

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 given 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.

While there may be scenarios where one homenet may wish to intentionally gain access through another, e.g. to share external connectivity costs, such scenarios are not discussed in this document.

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 into 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.

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 is desirable that appropriate borders can be configured to 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 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 LLN-based network, such as Smart Grid, 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, the exception being that it may not be possible to override policies defined by the ISP at the external border.

3.3.5. Configuration information from the ISP

In certain cases, it may be useful for the homenet to get certain configuration information from its ISP. For example, the homenet DHCP server may request and forward some options that it gets from its upstream DHCP server, though the specifics of the options may vary across deployments. There is potential complexity here of course should the homenet be multihomed.

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 document assumes that the guidance in the quoted text 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 stateless address autoconfiguration [RFC4862]), 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 Prefix Delegation (DHCP PD) [RFC3633], 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 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 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 6204 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. Where ULAs are used, it is preferable that only one /48 ULA covers the whole homenet, from which /64's can be delegated to the 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).

This document makes no specific recommendation towards solutions, 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

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.

The exposure of which traffic is sourced from the same homenet is thus similar to IPv4; the single IPv4 global address seen through use of IPv4 NAT gives the same hint as the global IPv6 prefix seen for IPv6 traffic.

While IPv4 NAT may obfuscate to an external observer which internal devices traffic is sourced from, IPv6, even with use of Privacy Addresses [RFC4941], adds additional exposure of which traffic is sourced from the same internal device, through use of the same IPv6 source address for a period of time.

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 based on a previously deployed protocol that has been shown to be reliable and robust, and that allows lightweight implementations. The availability of open source implementations is an important consideration. 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 types of physical interfaces must be accounted for in the homenet routed topology. Technologies such as Ethernet, WiFi, Multimedia over Coax Alliance (MoCA), etc. must be capable of coexisting in the same environment and should be treated as part of any routed deployment. The inclusion of physical 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 (this target is somewhat arbitrary, and was chosen based on how long a typical home user might wait before attempting another reset; ideally the routers might have some status light indicating they are converging, similar to an ADSL router light indicating it is establishing a connection to its ISP).

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 in 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 virtual machine (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.

[RFC4291] requires that any boundary of scope 4 or higher (i.e., admin-local or higher) be administratively configured. Thus the

boundary at the homenet-ISP border must be administratively configured, though that may be triggered by an administrative function such as DHCP-PD. Other multicast forwarding policy borders may also exist within the homenet, e.g., to/from a guest subnet, whilst the use of certain media types may also affect where specific multicast traffic is forwarded or routed.

There may be different drivers for multicast to be supported across the homenet, e.g., for homenet-wide service discovery should a multicast service discovery protocol of scope greater than link-local be defined, or potentially for multicast-based 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. As hinted above, it must be possible to scope or filter multicast traffic to avoid it being flooded to network media where devices cannot reasonably support it.

A homenet may not only use multicast internally, it may also be a consumer or provider of external multicast traffic, where the homenet's ISP supports such multicast operation. This may be valuable for example where live video applications are being sourced to/from the homenet.

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 PCP [RFC6887]. 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.

Considerations for differentiating neighbouring homenets are discussed in Section 3.3.1.

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 [RFC7084].

3.6.4. Exfiltration concerns

As homenets become more complex, with more devices, and with service discovery potentially enabled across the whole home, there are potential concerns over the leakage of information should devices use discovery protocols to gather information and report it to equipment vendors or application service providers.

While it is not clear how such exfiltration could be easily avoided, the threat should be recognised, be it from a new piece of hardware or some 'app' installed on a personal device.

3.6.5. 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.

3.6.6. 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 6204 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 graphical user interfaces (GUIs) 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 application programming interface (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 homenet 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 form 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 a new subnet is added to the homenet which already has a device of the same name. 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. 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 typically 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 authoritative name service provided by the ISP or perhaps an external third party.

Where zero configuration 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, where an attacker is able to insert a bogus DNS entry in the local cache, it is desirable to support appropriate name service security methods, including DNS Security Extensions (DNSSEC) [RFC4033], on both the authoritative server and the resolver sides. Where DNS is used, the homenet router or naming service must not prevent DNSSEC from operating.

While this document does not specify hardware requirements, it is worth noting briefly here that e.g., in support of DNSSEC,

appropriate homenet devices should have good random number generation capability, and future homenet specifications should indicate where high quality random number generators, i.e., with decent entropy, are needed.

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 externally provided service by some means.

3.7.4. 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. One approach is that the name space(s) used for the homenet would 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 externally provided 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. Any generated `<UniqueString>` should not be predictable; thus adding a salt/hash function would be desirable.

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. The IAB has issued a statement which explains why dotless domains should be considered harmful [IABdotless].

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.

It may be the case that if there are two or more CERs serving the home network, that if each has name space delegated from a different ISP there is the potential for devices in the home to have multiple fully qualified names under multiple domains.

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' (as described in [RFC6950] Section 4) is preferred for certain devices, whereby devices inside the homenet see different DNS responses to those outside.

Finally, 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. There are also some services, most notably email mail exchangers, where some operators have chosen to require a valid reverse lookup before accepting connections.

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)

[I-D.ietf-core-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 it may be desirable to convey any DNS domain search list that may be in effect for the homenet.

3.7.8. Devices roaming to/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. Devices may move between different homenets. In such cases it is desirable that devices may be accessed by the same name as is used in their 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 Quality of Service 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 Quality of Service 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 Quality of Service 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 overloading 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 have the general goal of being self-organising and configuring, and thus the network elements should not need to be proactively managed by the home user. Thus specific protocols that may be available to manage the network are not discussed in this document.

The network protocols used should, as far as possible, be able to self-configure, e.g. for prefixes to be assigned to router interfaces, or for devices to use zero-configuration protocols to discover services in the home. A home user would not be expected to, e.g., assign prefixes to links, or manage the DNS entries for the home network. Such expert operation should not be precluded, but it is not the norm.

There may still be some configuration parameters which are exposed to users, e.g., SSID name(s), or wireless security key(s). Users may also be expected to be aware of the functions of certain devices they connect, e.g., which are providing a server function, and they may be able to assign 'friendly names' to those devices, though service discovery protocols should make their selection as intuitive as possible.

As discussed in Section 3.6.1 the default setting on the homenet-ISP border for inbound traffic may be default deny, default allow, or some position inbetween. Whatever the default position, it should be possible for the user to change the setting.

Users may also be interested in the status of their networks and devices on the network, in which case simple self-monitoring mechanisms would be desirable. This may be particularly important when some fault arises in the network or with a device. 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 either model of management and monitoring is performed is out of scope of this document. It is expected that a separate document will follow to describe the operations and management model(s) for the types of home networks presented in this document.

A final consideration is that all network management and monitoring functions should be available over IPv6 transport, even where the homenet is dual-stack.

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 zero configuration 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. Security Considerations

Security considerations for the homenet architecture are discussed in Section 3.6 above.

6. IANA Considerations

This document has no actions for IANA.

7. References

7.1. Normative References

- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.
- [RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, December 2003.
- [RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, October 2005.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

7.2. 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.
- [RFC3002] Mitzel, D., "Overview of 2000 IAB Wireless Internetworking Workshop", RFC 3002, December 2000.

- [RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, January 2001.
- [RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, March 2005.
- [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.
- [RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.
- [RFC4864] Van de Velde, G., Hain, T., Droms, R., Carpenter, B., and E. Klein, "Local Network Protection for IPv6", RFC 4864, May 2007.
- [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.
- [RFC5890] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, August 2010.
- [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.
- [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.
- [RFC6887] Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", RFC 6887, April 2013.
- [RFC6950] Peterson, J., Kolkman, O., Tschofenig, H., and B. Aboba, "Architectural Considerations on Application Features in the DNS", RFC 6950, October 2013.
- [RFC7084] Singh, H., Beebee, W., Donley, C., and B. Stark, "Basic Requirements for IPv6 Customer Edge Routers", RFC 7084, November 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-06 (work in progress), February 2014.
- [I-D.ietf-ospf-ospfv3-autoconfig]

Lindem, A. and J. Arkko, "OSPFv3 Auto-Configuration", draft-ietf-ospf-ospfv3-autoconfig-06 (work in progress), February 2014.

[I-D.ietf-core-coap]

Shelby, Z., Hartke, K., and C. Bormann, "Constrained Application Protocol (CoAP)", draft-ietf-core-coap-18 (work in progress), June 2013.

[IABdotless]

"IAB Statement: Dotless Domains Considered Harmful", February 2013, <<http://www.iab.org/documents/correspondence-reports-documents/2013-2/iab-statement-dotless-domains-considered-harmful>>.

[Gettys11]

Gettys, J., "Bufferbloat: Dark Buffers in the Internet", March 2011, <<http://www.ietf.org/proceedings/80/slides/tsvarea-1.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, Elwyn Davies, 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 12

Changes made include:

- o Fixed minor typo nits introduced in -11.
- o Elwyn Davies' gen-art review comments addressed.
- o Some further IESG DISCUSS comments addressed.

B.2. Version 11 (after IESG review)

Changes made include:

- o Jouni Korhonen's OPSDIR review comments addressed.
- o Elwyn Davies' gen-art review comments addressed.
- o Considered secdir review by Samiel Weiler; many points addressed.
- o Considered APPSDIR review.
- o Addressed a large number of IESG comments and discusses.

B.3. Version 10 (after AD review)

Changes made include:

- o Minor changes/clarifications resulting from AD review

B.4. 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.5. 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.6. 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.7. 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.8. 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.9. 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.10. 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.11. 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: Standards Track
Expires: August 10, 2014

P. Pfister
B. Paterson
Cisco Systems
J. Arkko
Ericsson
February 6, 2014

Prefix and Address Assignment in a Home Network
draft-pfister-homenet-prefix-assignment-00

Abstract

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

Status of This Memo

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

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

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

This Internet-Draft will expire on August 10, 2014.

Copyright Notice

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

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

Table of Contents

1. Introduction	3
2. Requirements language	4
3. Prefix and Address Assignment Algorithms' Outline	4
4. Router Behavior	5
4.1. Data structures	5
4.2. Routers' Interfaces	7
4.3. Obtaining a Delegated Prefix	7
4.4. Designated Router	8
4.4.1. Sending Router Advertisement	8
4.4.2. Being the DHCP Server	8
4.5. Applying an Assignment on an Interface	9
4.6. DNS Support	10
5. Flooding Protocol Requirements	10
5.1. Router ID	10
5.2. Propagation Delay	10
5.3. Flooding Assigned Prefixes	11
5.4. Flooding Delegated Prefixes	11
5.5. Flooding Routers' Adresse Assignments	12
6. Prefix Assignment Algorithm	12
6.1. When to execute the Prefix Assignment Algorithm	12
6.2. Assignment Precedence	13
6.3. Testing Assignment's validity	13
6.4. Testing Assignment's availability	13
6.5. Accepting an Assigned Prefix	13
6.6. Making a New Assignment	14
6.7. Using Authoritative Prefix Assignments	15
6.8. Choosing the Assignment's Priority	15
6.9. Prefix Assignment Algorithm steps	16
7. Address Assignment Algorithm	17
7.1. Router's address pools	18
7.2. Address Assignment Algorithm	18
8. Hysteresis Principle	19
9. ULA and IPv4 Prefixes Generation	19
9.1. ULA Prefix Generation	19
9.2. IPv4 Private Prefix Generation	20
10. Manageability Considerations	20

11. Documents Constants	20
12. Security Considerations	21
13. References	21
13.1. Normative References	21
13.2. Informative References	22
Appendix A. Scarcity Avoidance Mechanism	23
A.1. Increasing Assigned Prefix Length	23
A.2. Foreseeing Prefixes Exhaustion	23
A.3. Cutting an Existing Assignment	24
Appendix B. Acknowledgments	24
Authors' Addresses	24

1. Introduction

This memo describes a fully distributed prefix and address assignment algorithm for home networks, running on top of any 'flooding protocol' that fulfills the specified requirements. It is expected that home border routers are allocated one or multiple IPv6 prefixes through DHCPv6 Prefix Delegation (PD) [RFC3633] or that prefixes are made available through other means. When an IPv4 address is assigned, a home private IPv4 prefix may be used with NAT to provide IPv4 connectivity to the whole home, as well as Unique Local Address prefixes [RFC4193] may be used in order to provide internal connectivity whenever global IPv6 connectivity is lost.

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

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

The rest of this memo is organized as follows. Section 2 defines the usual keywords, Section 3 outlines the algorithms functioning and features, Section 4 describes how a home router behaves when running the prefix and address assignment algorithm. Requirements for the underlying flooding protocol are detailed in Section 5. The prefix assignment algorithm is detailed in Section 6 and Section 7 focuses on the address assignment algorithm. Section 8 explains the hysteresis principles applied to both prefix and address assignments, Section 9 specifies the procedures for automatic generation of ULA

and IPv4 prefixes, Section 10 explains what administrative interfaces are useful for advanced users that wish to manually interact with the mechanisms, Section 12 discusses the security aspects and finally, Appendix A provides implementation guidelines for the optional scarcity avoidance mechanism.

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

2. Requirements language

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

3. Prefix and Address Assignment Algorithms' Outline

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

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

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

- o Each delegated prefix is efficiently subdivided so that each link is given a prefix for each available delegated prefix. If the

delegated prefix is too small given the size of the network, prefixes of arbitrary lengths may be used.

- o The algorithm is completely distributed. Routers may join and leave as well as Delegated Prefixes be added or deleted at any time.
- o IPv4 connectivity is provided whenever a home router gets an IPv4 address. To do so, a private IPv4 delegated prefix is generated and prefixes are assigned just like for IPv6.
- o The network may spontaneously generate and use a Unique Local Address (ULA) prefix.
- o Assignments are stable across reboots and some network changes (e.g. Adding or removing routers).
- o DHCP options like DNS servers, prefix colors, or any upcoming options may be attached to each prefixes and may be relayed down to the host when it is given addresses.
- o The user can manually assign prefixes to links. Such assignments will take precedence over automatically assigned prefixes.
- o Assignments and interfaces can be given priorities. When a delegated prefix is too small, such values may be used to prioritize prefix assignment to certain links.

4. Router Behavior

In a home network, all routers that want to participate in the prefix assignment algorithm **MUST** fulfill the requirements defined in this document. They **MUST** also use the same flooding protocol and routing protocol. The presence of an internal router that do not implement the flooding protocol and prefix assignment algorithm will not prevent the network from working as long as:

- o It doesn't act as a DHCP server on a link which is considered as internal by any other router.
- o It doesn't use any prefix that may be used by the prefix assignment algorithm.

4.1. Data structures

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

The router MUST maintain a list of all the Assigned Prefixes advertised by other routers. They are learnt through the mechanisms described in Section 5.3 and MUST contain the following information:

Prefix: The assigned prefix.

Router ID: The identifier of the advertising router.

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

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

Assignment's Priority: A value between PRIORITY_MIN and PRIORITY_MAX, quantifying the assignment's priority. The AP list is the result of the information provided by the flooding protocol, as specified in Section 5.3.

The router MUST maintain a list of all prefixes currently chosen to be applied on connected links. They are called Chosen Prefixes (CPs) and MUST contain the following information:

Prefix: The assigned prefix.

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

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

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

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

Applied: Whether that assignment is applied on link's configuration Section 4.5.

Chosen Prefixes that are marked as 'Advertised' are sent to other routers through the flooding protocol, and are therefore considered as Assigned Prefixes by other routers. The Prefix Assignment Algorithm goal is to make sure that all routers, on each link, select the same set of Chosen Prefixes.

The router MUST maintain a database of all its own address assignments, and address assignments made by other routers on connected links. The latter are learned through the mechanisms described in Section 5.5.

4.2. Routers' Interfaces

Each router's interface MUST either be considered as internal or external. Prefixes or addresses are only assigned to internal interfaces. The way an interface is selected as internal or external is out of the scope of this document.

If an internal interface's state is changed to external, all prefixes and addresses assigned on the considered interface MUST be deleted, and the prefix assignment algorithm MUST be run.

If an external interface's state is changed to internal, the prefix assignment algorithm MUST be run.

4.3. Obtaining a Delegated Prefix

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

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

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

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

- o A router MUST NOT advertise a prefix obtained through DHCPv6 PD if it doesn't understand the entirety of the provided options.

- o A router MUST NOT make or accept any assignment associated to a delegated prefix if it doesn't understand the entirety of the DHCP options advertised along-with the delegated prefix.

4.4. Designated Router

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

A router MUST consider itself as a designated router on a given link if one of the two following conditions is true:

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

Note: That particular order is motivated by the few cases where a router may voluntarily override an existing assignment by advertising an assignment of higher priority. In such a case, the designated router needs to remain the same.

4.4.1. Sending Router Advertisement

On a given link, the designated router MUST send router advertisements including Prefix Information Options for all the Chosen Prefixes associated to that link. SLAAC MAY be enabled depending on the router's configuration and assignments prefix length. The valid and preferred lifetimes MUST be set to values lower or equal to the associated Delegated Prefix's valid and preferred lifetimes.

4.4.2. Being the DHCP Server

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

behaving as a DHCP server whenever it is not the link's designated router anymore.

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

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

4.5. Applying an Assignment on an Interface

Once a Chosen Prefix is created, a router first waits some time in order to detect possible collisions (Section 8). Once the timeout is elapsed and no collision is detected, the prefix is applied by executing the following steps:

- o The router updates its interface configuration so that the prefix is assigned to the considered link.
- o The router updates the routing protocol configuration so that it starts advertising the prefix. Depending on the implementation, this step may not be needed as the routing protocol directly gets its configuration information from the interfaces configuration.
- o If necessary, the router starts selecting an address for itself as defined in Section 7.
- o If the router is the designated router on the considered link, it starts sending the Prefix Information Option with the considered prefix, as specified in Section 4.4.1.
- o If the router is the designated router on the considered link, it starts behaving as a DHCP server, as defined in Section 4.4.2, for the considered assigned prefix.

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

4.6. DNS Support

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

Whenever the router knows which DNS server to use, or is acting as a DNS relay, it SHOULD include DNS DHCPv6 option ([RFC3646]) along-with host's configuration messages and include the Router Advertisement DNS options ([RFC6106]) when sending RAs.

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

5. Flooding Protocol Requirements

In this document, the Flooding Protocol (FP) refers to a protocol enabling information propagation to the whole network. It was not specified in order to allow the working group to independently decide which routing protocol, configuration protocol, and prefix assignment method to use within the home network. Routing protocol, like OSPF or ISIS, could be extended in order to fulfill the requirements, as well as new dedicated and optimized protocols could be proposed.

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

5.1. Router ID

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

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

5.2. Propagation Delay

The FP MUST provide an approximate upper bound of the time it takes for an update to be propagated to the whole network. This value is referred to as the FLOODING_DELAY. The algorithm ensures that, as

long as the upper bound is respected, two identical prefixes will never be applied to different links, and two different prefixes will never be applied to the same link. The algorithm and the network will recover when the upper bound is exceeded, but collisions may appear in the routing protocol and errors may be propagated to upper layers.

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

5.3. Flooding Assigned Prefixes

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

5.4. Flooding Delegated Prefixes

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

Prefix: The delegated prefix.

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

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

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

DHCP information: DHCPv6 encoded options attached to the delegated prefix.

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

5.5. Flooding Routers' Adresse Assignments

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

6. Prefix Assignment Algorithm

The Prefix Assignment Algorithm is a distributed algorithm that assigns one prefix from each available Delegated Prefix on every link that is considered as internal by at least one connected router. The algorithm itself makes no difference whether the delegated prefix is global IPv6, ULA or IPv4. IPv4 prefixes are written in their IPv4-mapped IPv6 form, as defined in [RFC4291] (i.e. ::ffff:A.B.C.D/X with X >= 96).

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

6.1. When to execute the Prefix Assignment Algorithm

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

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

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

6.2. Assignment Precedence

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

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

6.3. Testing Assignment's validity

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

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

6.4. Testing Assignment's availability

A prefix is said to be available if it is not included and does not include any other assignment made by any router in the network.

6.5. Accepting an Assigned Prefix

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

- o The same prefix.
- o The same link ID.
- o The authoritative bit set to false.
- o The same priority.
- o The advertised bit value set as specified by the algorithm.

- o The applied bit is unset. It is set when the timer elapsed if the entry still exists.

6.6. Making a New Assignment

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

If $X < 64$: Prefix length = 64

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

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

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

When the algorithm decides to make a new assignment, it looks in the stable storage for an available assignment that was previously applied on the current interface and that is included in the current delegated prefix. If no available assignment can be found this way, the new prefix MUST be randomly selected among prefixes in the current Delegated Prefix that are still available. Implementing a uniform selection among all available prefixes may be challenging, but an implementation SHOULD at least be able to make an exhaustive search when the address space is small, and make multiple tentatives when the address space is too big.

If no available prefix is found, the assignment fails. If implemented, the router MAY decide to execute the Prefix Scarcity Avoidance mechanisms, as proposed in Appendix A.

When a new assignment is made, a new Chosen Prefix entry is created.

- o The prefix value is set to the chosen prefix.
- o The link ID is the ID of the link on which the assignment is made.
- o The authoritative bit is set to false.
- o The priority is set to a value between PRIORITY_AUTO_MIN and PRIORITY_AUTO_MAX (Section 6.8).
- o The advertised bit is set.

- o The applied bit is unset. It is set when the timer elapsed if the entry still exists.

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

6.7. Using Authoritative Prefix Assignments

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

There are two kinds of Authoritative Prefix Assignments.

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

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

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

6.8. Choosing the Assignment's Priority

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

- o PRIORITY_DEFAULT SHOULD be used as default.

- o Other values between `PRIORITY_AUTO_MIN` and `PRIORITY_AUTO_MAX` MAY be dynamically chosen by the implementation.
- o Other values between `PRIORITY_AUTHORITY_MIN` and `PRIORITY_AUTHORITY_MAX` MUST NOT be used if not stated by an authority (by static or dynamic configuration).
- o Other values are reserved.

6.9. Prefix Assignment Algorithm steps

In this section are detailed the steps of the Prefix Assignment Algorithm.

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

The following steps are then executed for every combination of delegated prefix and interface.

- o If the current interface is external, ignore that interface.
- o If the delegated prefix strictly contains another delegated prefix, ignore that delegated prefix.
- o If the delegated prefix is equal to an already considered delegated prefix, ignore that delegated prefix.
- o Look for a valid Assigned Prefix, advertised by another router on the current interface and included in the current Delegated Prefix.
- o Look for a Chosen Prefix associated to the current interface and included in the current Delegated Prefix.
- o There are four possibilities at this stage.
 1. If no AP is found, and no CP is found, a new assignment MUST be made if and only if the router considers itself as the designated router. See Section 6.6.
 2. If an AP is found, and no CP is found, the AP MUST be accepted. The new CP's advertised bit MUST be set if and only if the router considers itself as the designated router.
 3. If no AP is found, and a CP is found, the router MUST check if the CP's assignment is valid. If it is, the local assignment

is marked as valid and advertised. If it isn't, it is destroyed and the algorithm applies case 1.

4. If both an AP and a CP are found, the router must check if the prefixes are the same. If they are different and if the CP's Authoritative bit is not set, the CP MUST be deleted and the algorithm applies case 2. If the prefixes are the same, the CP must be updated with the AP's priority value, marked as valid, and advertised if and only if the router considers itself as designated on the link.

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

7. Address Assignment Algorithm

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

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

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

7.1. Router's address pools

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

If $X < 64$: SLAAC MUST be used

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

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

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

7.2. Address Assignment Algorithm

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

Routers MUST store applied address assignments in stable storage and reuse the same addresses whenever possible. At least the five previously applied addresses should be stored.

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

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

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

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

- o The associated Chosen Prefix is deleted or moved to another link.

- o Some other router, with an higher router ID, is advertising the same address on the same link.

8. Hysteresis Principle

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

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

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

9. ULA and IPv4 Prefixes Generation

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

9.1. ULA Prefix Generation

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

- o It is the network leader.
- o No other ULA delegated prefix is advertised by any other router.

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

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

9.2. IPv4 Private Prefix Generation

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

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

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

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

10. Manageability Considerations

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

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

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

11. Documents Constants

PRIORITY_MIN	0
PRIORITY_AUTHORITY_MIN	4
PRIORITY_AUTO_MIN	6
PRIORITY_DEFAULT	8
PRIORITY_AUTO_MAX	10
PRIORITY_AUTHORITY_MAX	12
PRIORITY_MAX	15

12. Security Considerations

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

1. The flooding protocol is not protected.
2. The flooding protocol's protection is binary: An allowed router may send any type of packets in the name of other routers.
3. All advertised messages are individually signed by the sender.

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

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

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

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

13. References

13.1. Normative References

- [RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

- [RFC3203] T'Joens, Y., Hublet, C., and P. De Schrijver, "DHCP reconfigure extension", RFC 3203, December 2001.
- [RFC3646] Droms, R., "DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3646, December 2003.
- [RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, October 2005.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.
- [RFC6106] Jeong, J., Park, S., Beloeil, L., and S. Madanapalli, "IPv6 Router Advertisement Options for DNS Configuration", RFC 6106, November 2010.
- [RFC6603] Korhonen, J., Savolainen, T., Krishnan, S., and O. Troan, "Prefix Exclude Option for DHCPv6-based Prefix Delegation", RFC 6603, May 2012.

13.2. Informative References

- [RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, December 2003.
- [I-D.ietf-homenet-arch]
Chown, T., Arkko, J., Brandt, A., Troan, O., and J. Weil, "IPv6 Home Networking Architecture Principles", draft-ietf-homenet-arch-11 (work in progress), October 2013.
- [I-D.dimitri-zospf]
Dimitrelis, A. and A. Williams, "Autoconfiguration of routers using a link state routing protocol", draft-dimitri-zospf-00 (work in progress), October 2002.
- [I-D.chelius-router-autoconf]
Chelius, G., Fleury, E., and L. Toutain, "Using OSPFv3 for IPv6 router autoconfiguration", draft-chelius-router-autoconf-00 (work in progress), June 2002.

[I-D.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.

Appendix A. Scarcity Avoidance Mechanism

When not enough addresses are available, a router may decide to execute procedures intended to avoid prefix scarcity. Different approaches are possible. This section intends to provide guidelines for such procedures implementation. They are optional and are compatible with routers that only support basic requirements defined in this document.

A.1. Increasing Assigned Prefix Length

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

A.2. Foreseeing Prefixes Exhaustion

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

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

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

A.3. Cutting an Existing Assignment

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

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

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

Appendix B. Acknowledgments

This document is the continuation of the work being done in [I-D.arkko-homenet-prefix-assignment]. The authors would like to thank all the people that participated in the previous document's development as well as the present one. In particular, the authors would like to thank to Tim Chown, Fred Baker, Mark Townsley, Lorenzo Colitti, Ole Troan, Ray Bellis, Markus Stenberg, Wassim Haddad, Joel Halpern, Samita Chakrabarti, Michael Richardson, Anders Brandt, Erik Nordmark, Laurent Toutain, Ralph Droms, Acee Lindem and Steven Barth for interesting discussions in this problem space. The authors would also like to point out some past work in this space, such as those in [I-D.chelius-router-autoconf] or [I-D.dimitri-zospf].

Authors' Addresses

Pierre Pfister
Cisco Systems
Paris
France

Email: pierre@darou.fr

Benjamin Paterson
Cisco Systems
Paris
France

Email: benjamin@paterson.fr

Jari Arkko
Ericsson
Jorvas 02420
Finland

Email: jari.arkko@piuha.net

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 09, 2014

M. Stenberg
February 05, 2014

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

Abstract

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

Status of This Memo

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

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

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

This Internet-Draft will expire on August 09, 2014.

Copyright Notice

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

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

Table of Contents

1. Introduction	2
2. Requirements language	3
3. Hybrid proxy - what to configure	3
3.1. Conflict resolution within network	4
3.2. Per-link DNS-SD forward zone names	4
3.3. Reasonable defaults	4
3.3.1. Network-wide unique link name (scheme 1)	5
3.3.2. Node name (scheme 2)	5
3.3.3. Link name (scheme 2)	5
4. TLVs	5
4.1. DNS Delegated Zone TLV	5
4.2. Domain Name TLV	6
4.3. Node Name TLV	7
5. Desirable behavior	7
5.1. DNS search path in DHCP requests	7
5.2. Hybrid proxy	7
5.3. Hybrid proxy network zeroconf daemon	8
6. Security Considerations	8
7. References	8
7.1. Normative references	8
7.2. Informative references	9
Appendix A. Example configuration	9
A.1. Topology	9
A.2. OSPFv3-DNS interaction	9
A.3. TLV state	10
A.4. DNS zone	11
A.5. Interaction with hosts	12
Appendix B. Implementation	12
Appendix C. Why not just proxy Multicast DNS?	12
C.1. General problems	12
C.2. Stateless proxying problems	13
C.3. Stateful proxying problems	13
Appendix D. Acknowledgements	14
Author's Address	14

1. Introduction

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

This document describes what sort of configuration the participating hybrid proxy servers require, as well as how it can be provided using any network-level state sharing mechanism (such as routing protocol)

and a naming scheme which does not even need to be same across the whole covered network to work (given working conflict resolution does work). The scheme can be used to provision both forward and reverse DNS zones which employ hybrid proxy for heavy lifting.

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

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

2. Requirements language

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

3. Hybrid proxy - what to configure

Beyond the low-level translation mechanism between unicast and multicast service discovery, the hybrid proxy draft [I-D.cheshire-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 some other protocol's presence (or lack of it) on a link to determine internal links within the covered network, and some other signs (depending on the deployment) such as DHCPv6 Prefix Delegation (as described in [RFC3633]) to determine external links that should not be covered.

For each covered link we want forward DNS zone delegation to an appropriate node which is connected to a link, and running hybrid proxy. Therefore the links' forward DNS zone names should be unique across the network. We also want to populate reverse DNS zone similarly for each IPv4 or IPv6 prefix in use.

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

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

3.1. Conflict resolution within network

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

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

3.2. Per-link DNS-SD forward zone names

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

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

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

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

3.3. Reasonable defaults

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

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

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

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

3.3.2. Node name (scheme 2)

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

3.3.3. Link name (scheme 2)

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

4. TLVs

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

4.1. DNS Delegated Zone TLV

This TLV is effectively a combined NS and A/AAAA record for a zone. It MUST be supported by implementations conforming to this specification. Implementations SHOULD provide forward zone per link (or optimizing a bit, zone per link with Multicast DNS traffic). Implementations MAY provide reverse zone per prefix using this same mechanism. If multiple nodes advertise same reverse zone, it should

be assumed that they all have access to the link with that prefix. However, as noted in Section 5.3, mainly only the node with highest precedence on the link should publish this TLV.

Contents:

Address field is IPv6 address (e.g. 2001:db8::3) or IPv4 address mapped to IPv6 address (e.g. ::FFFF:192.0.2.1) where the authoritative DNS server for Zone can be found. If the address field is all zeros, the Zone is under global DNS hierarchy and can be found using normal recursive name lookup starting at the authoritative root servers (This is mostly relevant with the S bit below).

S bit indicates that this delegated zone consists of a full DNS-SD domain, which should be used as base for DNS-SD domain enumeration (that is, (field)._dns-sd._udp.(zone) exists). Forward zones MAY have this set. Reverse zones MUST NOT have this set. This can be used to provision DNS search path to hosts for non-local services (such as those provided by ISP, or other manually configured service providers).

B bit indicates that this delegated zone should be included in network's DNS-SD browse list of domains at b._dns-sd._udp.(domain). Local forward zones SHOULD have this set. Reverse zones SHOULD NOT have this set.

Zone is the label sequence of the zone, encoded according to section 3.1. ("Name space definitions") of [RFC1035]. Note that name compression is not required here (and would not have any point in any case), as we encode the zones one by one. The zone MUST end with empty label.

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

4.2. Domain Name TLV

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

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

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

4.3. Node Name TLV

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

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

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

5. Desirable behavior

5.1. DNS search path in DHCP requests

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

5.2. Hybrid proxy

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

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

5.3. Hybrid proxy network zeroconf daemon

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

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

6. Security Considerations

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

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

7. References

7.1. Normative references

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

7.2. Informative references

[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:

```

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

```

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

The user-visible set of links are [3] and [4]; each consisting of a LAN and WLAN link. We assume that ISP provides 2001:db8::/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).(node).(domain), and no configuration whatsoever, here's what

happens (the steps are interleaved in practise but illustrated here in order):

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

A.3. TLV state

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

(from CER)

DZ {s=1, zone="isp.example.com"}

(from IR1)

NN {name="node"}

DZ {address=2001:db8:1:1::1, b=1,
zone="lan.node.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.node.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)
NN {name="node-2"}

DZ {address=2001:db8:2:1::1, b=1,
    zone="lan.node-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.node-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 nodes, just based on dumping the delegated zone TLVs as NS+AAAA records, and optionally domain list browse entry for DNS-SD:

```

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

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

node AAAA 2001:db8:1:1::1
node-2 AAAA 2001:db8:2:1::1

node NS node
node-2 NS node-2

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

```

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

A.5. Interaction with hosts

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

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

Appendix B. Implementation

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

Appendix C. Why not just proxy Multicast DNS?

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

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

C.1. General problems

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

First reason is the conflict resolution depends on 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

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 09, 2014

M. Stenberg

S. Barth

February 05, 2014

Home Networking Control Protocol
draft-stenberg-homenet-hncp-00

Abstract

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

Status of This Memo

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

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

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

This Internet-Draft will expire on August 09, 2014.

Copyright Notice

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

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

Table of Contents

1. Introduction	3
2. Requirements language	3
3. Data model	3
4. Operation	4
4.1. Trickle-Driven Status Updates	4
4.2. Protocol Messages	5
4.2.1. Network State Update (NetState)	5
4.2.2. Network State Request, (NetState-Req)	5
4.2.3. Node Data Request (Node-Req)	6
4.2.4. Network and Node State Reply (NetNode-Reply)	6
4.3. HNCP Protocol Message Processing	6
4.4. Adding and Removing Neighbors	8
4.5. Purging Unreachable Nodes	8
5. Type-Length-Value objects	8
5.1. Request TLVs (for use within unicast requests)	9
5.1.1. Request Network State TLV	9
5.1.2. Request Node Data TLV	9
5.2. Data TLVs (for use in both multi- and unicast data)	10
5.2.1. Node Link TLV	10
5.2.2. Network State TLV	10
5.2.3. Node State TLV	10
5.2.4. Node Data TLV	11
5.2.5. Node Public Key TLV (within Node Data TLV)	11
5.2.6. Neighbor TLV (within Node Data TLV)	12
5.3. Custom TLV (within/without Node Data TLV)	12
5.4. Authentication TLVs	13
5.4.1. Certificate-related TLVs	13
5.4.2. Signature TLV	13
6. Border Discovery and Prefix Assignment	13
7. DNS-based Service Discovery	17
7.1. DNS Delegated Zone TLV	18
7.2. Domain Name TLV	18
7.3. Router Name TLV	18
8. Routing support	18
8.1. Protocol Requirements	18
8.2. Announcement	19
8.3. Protocol Selection	20
8.4. Fallback Mechanism	20
9. Security Considerations	21
10. IANA Considerations	22
11. References	23
11.1. Normative references	23
11.2. Informative references	23
Appendix A. Some Outstanding Issues	25
Appendix B. Some Obvious Questions and Answers	25

Appendix C. Draft source	26
Appendix D. Acknowledgements	26
Authors' Addresses	27

1. Introduction

HNCP is designed to synchronize state across a Homenet (or other small site) in order to facilitate automated configuration within the site, integration with trusted bootstrapping [I-D.behringer-homenet-trust-bootstrap] and default perimeter detection [I-D.kline-homenet-default-perimeter], automatic IP prefix distribution [I-D.pfister-homenet-prefix-assignment], and service discovery across multiple links within the homenet as defined in [I-D.stenberg-homenet-dnssd-hybrid-proxy-network-zeroconf].

HNCP is designed to provide enough information for a routing protocol to operate without homenet-specific extensions. In homenet environments where multiple IPv6 prefixes are present, routing based on source and destination address is necessary [I-D.troan-homenet-sadr]. Routing protocol requirements for source and destination routing are described in section 3 of [I-D.baker-rtgwg-src-dst-routing-use-cases].

A GPLv2-licensed implementation of the HNCP protocol is currently under development at <https://github.com/sbyx/hnetd/> [1]. Comments and/or pull requests are welcome.

An earlier implementation using many of the same principles, algorithms and data structures built within OSPFv3 is available at <http://www.homewrt.org/doku.php?id=downloads> [2].

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. Data model

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

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

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

Latest update sequence number. A four octet number that is incremented anytime TLV data changes are detected.

Relative time, in milliseconds, since last publishing of the current TLV data set.

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

4. Operation

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

4.1. Trickle-Driven Status Updates

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

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

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

Imax MUST be at least one minute.

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

4.2. Protocol Messages

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

In order to facilitate fast comparing of local state with that in a received message update, all TLVs in every encoding scope (either root level, within the message itself, or within a container TLV) MUST be placed in ascending order based on the binary comparison of both TLV header and value. By design, the TLVs which MUST be present have the lowest available type values, ensuring they will naturally occur at the start of the Protocol Message, resembling a fixed format preamble.

4.2.1. Network State Update (NetState)

This Message SHOULD be sent as a multicast message.

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

Node Link TLV (Section 5.2.1).

Network State TLV (Section 5.2.2).

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

If HNCP security is enabled, authentication TLVs (Section 5.4) MUST be present.

4.2.2. Network State Request, (NetState-Req)

This Message MUST be sent as a unicast message.

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

Node Link TLV (Section 5.2.1).

Request Network State TLV (Section 5.1.1).

If HNCP security is enabled, authentication TLVs (Section 5.4) MUST be present.

4.2.3. Node Data Request (Node-Req)

This Message MUST be sent as a unicast message.

MUST be present:

Node Link TLV (Section 5.2.1).

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

If HNCP security is enabled, authentication TLVs (Section 5.4) MUST be present.

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

This Message MUST be sent as a unicast message.

MUST be present:

Node Link TLV (Section 5.2.1).

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

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

If HNCP security is enabled, authentication TLVs (Section 5.4) MUST be present.

4.3. HNCP Protocol Message Processing

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

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

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

Upon receipt of:

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

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

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

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

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

If a message containing Node State TLVs (Section 5.2.3) is received via unicast or multicast with the node's own node identifier and a higher update number than current local value, or the same update number and different hash, there is an error somewhere. A recommended default way to handle this is to attempt to assert local state by increasing the local update number to a value higher than that received and republish node data using the same node identifier. If this happens more than 3 times in 60 seconds and the local node identifier is not globally unique, there may be more than one router with the same node identifier on the network. If HNCP security is not enabled, a new node identifier SHOULD be generated and node data

republished accordingly. If HNCP security is enabled, this is event is highly unlikely to occur as collision of identifier hashes for public keys is highly unlikely.

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

4.4. Adding and Removing Neighbors

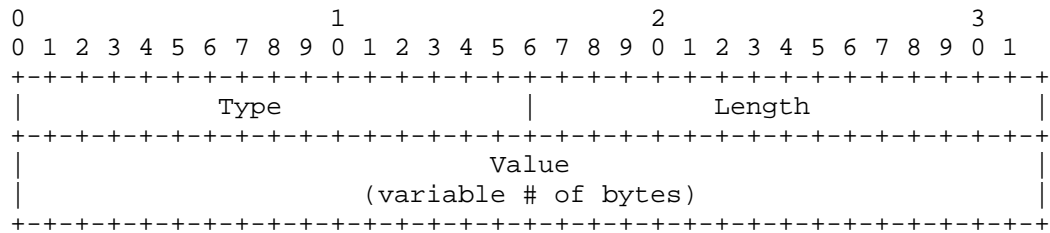
Whenever multicast message or unicast reply is received on a link from another node, the node should be added as Neighbor TLV (Section 5.2.6) for current node. If nothing (for example - no router advertisements, no HNCP traffic) is received from that neighbor in I_{max} seconds and the neighbor is not in neighbor discovery cache (and L2 indication of presence is available), at least 3 attempts to ping it with request network state message (Section 4.2.2) SHOULD be sent with increasing timeouts (e.g. 1, 2, 4 seconds). If even after suitable period after the last message nothing is received, the Neighbor TLV MUST be removed so that there are no dangling neighbors. As an alternative, if there is a layer 2 unreachability notification of some sort available for either whole link or for individual neighbor, it MAY be used to immediately trigger removal of corresponding Neighbor TLV(s).

4.5. Purging Unreachable Nodes

Nodes should be purged when unreachable. When node data has changed, the neighbor graph SHOULD be traversed for each node following the Neighbor TLVs, purging nodes that were found entirely unreachable (not traversed).

5. Type-Length-Value objects

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



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

Notation:

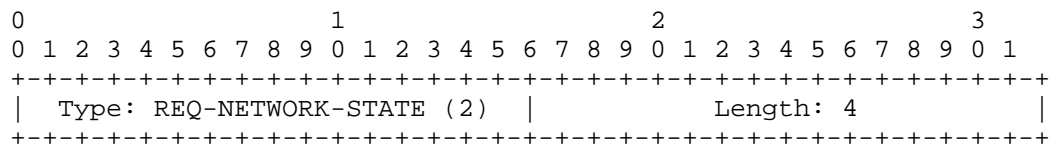
.. = octet string concatenation operation

H(x) = MD5 hash of x

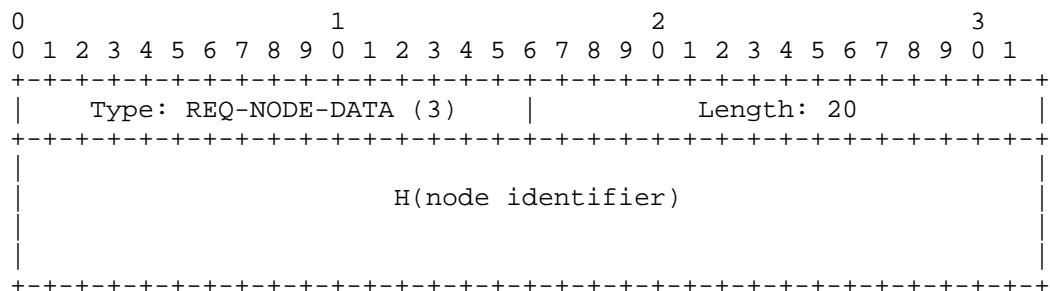
H-64(x) = H(x) truncated by taking just first 64 bits of the result.

5.1. Request TLVs (for use within unicast requests)

5.1.1. Request Network State TLV

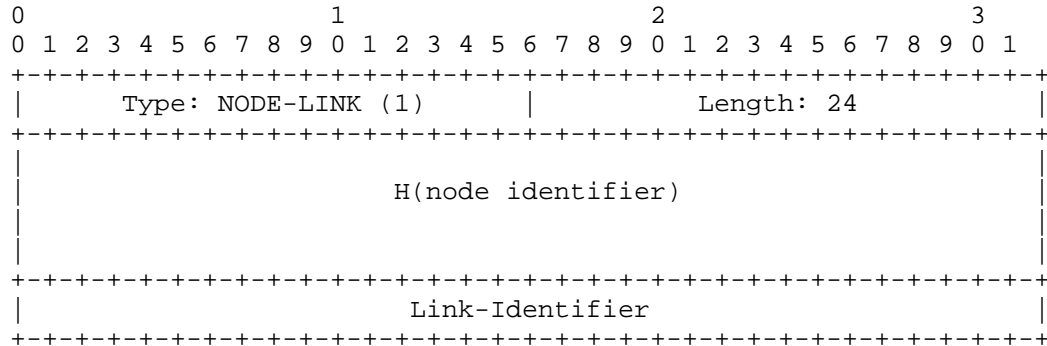


5.1.2. Request Node Data TLV

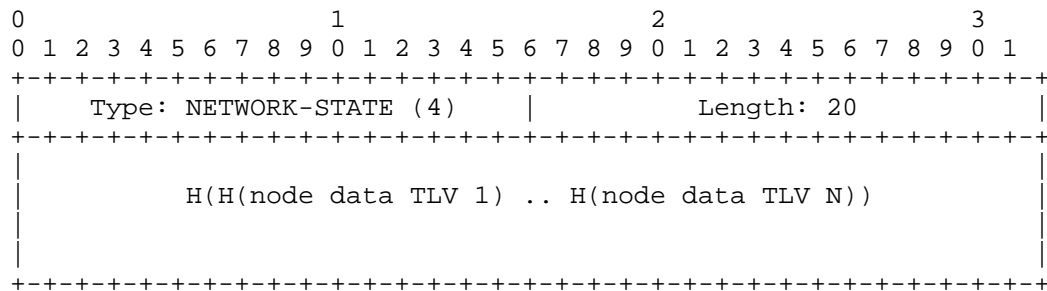


5.2. Data TLVs (for use in both multi- and unicast data)

5.2.1. Node Link TLV

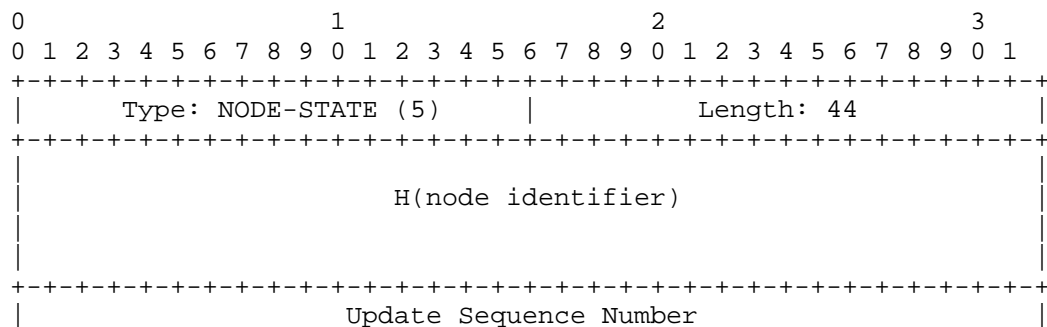


5.2.2. Network State TLV



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

5.2.3. Node State TLV



```

+-----+
|                               |
|      Milliseconds since Origination      |
|-----+-----+-----+-----+-----+-----+-----+-----+
|                               |
|                               |
|                               |
|                               |
|                               |
|                               |
|                               |
|                               |
|-----+-----+-----+-----+-----+-----+-----+-----+
|                               |
|      H(node data TLV)      |
|-----+-----+-----+-----+-----+-----+-----+-----+

```

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

5.2.4. Node Data TLV

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|      Type: NODE-DATA (6)      |      Length: >= 24      |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                               |
|                               |
|                               |
|                               |
|                               |
|                               |
|                               |
|                               |
|-----+-----+-----+-----+-----+-----+-----+-----+
|                               |
|      Update Sequence Number      |
|-----+-----+-----+-----+-----+-----+-----+-----+
|                               |
|      Nested TLVs containing node information      |
|-----+-----+-----+-----+-----+-----+-----+-----+

```

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

If signatures are in use, the Node Data TLV SHOULD also contain the originator's own Signature TLV (Section 5.4.2).

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

```

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

```

Public Key (raw node identifier)

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

5.2.6. Neighbor TLV (within Node Data TLV)

[illegible]

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

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

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

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


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

or

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

5.4. Authentication TLVs

5.4.1. Certificate-related TLVs

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

5.4.2. Signature TLV

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

6. Border Discovery and Prefix Assignment

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

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

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

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

An interface MUST be considered external if at least one of the following conditions is satisfied:

1. The interface has a fixed category classifying it as external.
2. A delegated prefix could be acquired by running a DHCPv6-client on the interface.
3. An IPv4-address could be acquired by running a DHCP-client on the interface.
4. HNCP security is enabled and there are routers on the interface which could not be authenticated.

Each router MUST continuously scan each active interface that does not have a fixed category in order to dynamically reclassify it if necessary. The router therefore runs an appropriately configured DHCP and DHCPv6-client as long as the interface is active including states where it considers the interface to be internal. The router SHOULD wait for a reasonable time period (5 seconds as a possible default) in which the DHCP-clients can acquire a lease before treating a newly activated or previously external interface as internal. Once it treats a certain interface as internal it MUST start forwarding traffic with appropriate source addresses between its internal interfaces and allow internal traffic to reach external networks. Once a router detects an interface to be external it MUST stop any previously enabled internal forwarding. In addition it SHOULD announce the acquired information for use in the homenet as described in later sections of this draft if the interface appears to be connected to an external network.

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

```

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

```

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

```

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

```

and

```

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

```

Each delegated prefix is encoded using one TLV inside an EXTERNAL-CONNECTION TLV. For external IPv4 connections the prefix is encoded in the form of an IPv4-mapped address [RFC4291] and is usually from a private address range [RFC1597]. The related TLV is defined as follows.

```

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

```

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

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

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

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

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

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

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

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|  Type: ASSIGNED-PREFIX (43)  |          Length: >= 9          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               Link Identifier                    |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|  R. |A| Pref. | Prefix Length |                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               Prefix Address                    |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

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

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

7.1. DNS Delegated Zone TLV

```

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

```

7.2. Domain Name TLV

```

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

```

7.3. Router Name TLV

```

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

```

8. Routing support

8.1. Protocol Requirements

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

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

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

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

8.2. Announcement

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

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

```

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

```

Protocol ID is one of:

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

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

8.3. Protocol Selection

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

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

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

8.4. Fallback Mechanism

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

1. Interpret the neighbour information received via HNCP as a graph of connected routers.
2. Use breadth-first traversal to determine the next-hop and hop-count in the path to each router in the homenet:

Start the traversal with the immediate neighbours of the router running the algorithm.

Always visit the immediate neighbours of a router in ascending order of their router ID.

Never visit a router more often than once.

3. For each delegated prefix P of any router R in the homenet:
Create a default route via the next-hop for R acquired in #2.

Each such route MUST be source-restricted to only apply to traffic with a source address within P and its metric MUST reflect the hop-count to R.

4. For each assigned prefix A of a router R: Create a route to A via the next-hop for R acquired in #2. Each such route MUST NOT be source-restricted.
5. For the first router R visited in the traversal announcing an IPv4-uplink: Create a default IPv4-route via the next-hop for R acquired in #2.
6. For each assigned IPv4-prefix A of a router R: Create an IPv4-route to A via the next-hop for R acquired in #2.

9. Security Considerations

General security issues for Home Networks are discussed at length in [I-D.ietf-homenet-arch]. The protocols used to setup IP in home networks today have very little security enabled within the control protocol itself. For example, DHCP has defined [RFC3118] to authenticate DHCP messages, but this is very rarely implemented in large or small networks. Further, while PPP can provide secure authentication of both sides of a point to point link, it is most often deployed with one-way authentication of the subscriber to the ISP, not the ISP to the subscriber. HNCP aims to make security as easy as possible for the implementer by including built-in capabilities for authentication of node data being exchanged as well as the protocol messages themselves, but it is ultimately up to the shipping system to take advantage of the protocol constructs defined.

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

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

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

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

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

10. IANA Considerations

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

0: Reserved (should not happen on wire)

1: Node link

2: Request network state

3: Request node data

4: Network state

5: Node state

6: Node data

7: Node public key

8: Neighbor

9: Custom

41: External connection

42: Delegated prefix

43: Assigned prefix

44: DHCP-data
45: DHCPv6-data
46: Router-address
50: DNS Delegated Zone
51: Domain name
52: Node name
60: Routing protocol
65535: Signature

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

11. References

11.1. Normative references

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC6206] Levis, P., Clausen, T., Hui, J., Gnawali, O., and J. Ko, "The Trickle Algorithm", RFC 6206, March 2011.
- [I-D.pfister-homenet-prefix-assignment]
Pfister, P., Arkko, J., and B. Paterson, "Prefix and Address Assignment in a Home Network", draft-pfister-homenet-prefix-assignment-00 (work in progress), January 2014.
- [I-D.stenberg-homenet-dnssd-hybrid-proxy-network-zeroconf]
Stenberg, M., "Auto-Configuration of a Network of Hybrid Unicast/Multicast DNS-Based Service Discovery Proxy Nodes", draft-pfister-homenet-prefix-assignment-00 (work in progress), January 2014.

11.2. Informative references

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

- [RFC3004] Stump, G., Droms, R., Gu, Y., Vyaghrapuri, R., Demirtjis, A., Beser, B., and J. Privat, "The User Class Option for DHCP", RFC 3004, November 2000.
- [RFC3118] Droms, R. and W. Arbaugh, "Authentication for DHCP Messages", RFC 3118, June 2001.
- [RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.
- [RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.
- [RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, December 2003.
- [RFC1597] Rekhter, Y., Moskowitz, R., Karrenberg, D., and G. de Groot, "Address Allocation for Private Internets", RFC 1597, March 1994.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.
- [I-D.ietf-homenet-arch]
Chown, T., Arkko, J., Brandt, A., Troan, O., and J. Weil,
"IPv6 Home Networking Architecture Principles", draft-
ietf-homenet-arch-11 (work in progress), October 2013.
- [I-D.troan-homenet-sadr]
Troan, O. and L. Colitti, "IPv6 Multihoming with Source
Address Dependent Routing (SADR)", draft-troan-homenet-
sadr-01 (work in progress), September 2013.
- [I-D.behringer-homenet-trust-bootstrap]
Behringer, M., Pritikin, M., and S. Bjarnason,
"Bootstrapping Trust on a Homenet", draft-behringer-
homenet-trust-bootstrap-00 (work in progress), October
2012.
- [I-D.baker-rtgwg-src-dst-routing-use-cases]
Baker, F., "Requirements and Use Cases for Source/
Destination Routing", draft-baker-rtgwg-src-dst-routing-
use-cases-00 (work in progress), August 2013.
- [I-D.kline-homenet-default-perimeter]

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

[I-D.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.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.

[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.

Appendix A. Some Outstanding Issues

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

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

Consider if using Trickle with $k=1$ really pays off, as we need to do reachability checks if L2 doesn't provide them periodically in any case. Using Trickle with $k=\text{inf}$ would remove the need for unicast reachability checks, but at cost of extra multicast traffic. On the other hand, $N*(N-1)/2$ unicast reachability checks when lot of routers share a link is not appealing either.

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

Appendix B. Some Obvious Questions and Answers

Q: Why not use TCP?

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

Q: Why effectively build a link state routing protocol without routing?

A: It felt like a good idea at the time. It does not require periodic flooding except for very minimal Trickle-based per-link state maintenance (potentially also neighbor reachability checks if so desired).

Q: Why not multicast-only?

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

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

A: Scalability of protocol isn't really affected by using real (=cryptographic) hash function.

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

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

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

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

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

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

Appendix C. Draft source

As usual, this draft is available at <https://github.com/fingon/ietf-drafts/> [3] in source format (with nice Makefile too). Feel free to send comments and/or pull requests if and when you have changes to it!

Appendix D. Acknowledgements

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

Authors' Addresses

Markus Stenberg
Helsinki 00930
Finland

Email: markus.stenberg@iki.fi

Steven Barth

Email: cyrus@openwrt.org

Homenet
Internet-Draft
Intended status: Informational
Expires: August 16, 2014

T. Winters, Ed.
UNH-IOL
February 14, 2014

Service Provider Edge Router Interaction
draft-winters-homenet-sper-interaction-01

Abstract

This document describes the interaction between a Service Provider Gateway fixed at the home edge, and the Home Networking interior routers. It assesses the interactions between existing routers implementing [RFC7084] and the Home Networking routers. The document will also define the interactions between other Service Provider Edge Router (eg. HIPnet) and Home Networking router.

Status of this Memo

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

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

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

This Internet-Draft will expire on August 16, 2014.

Copyright Notice

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

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

Table of Contents

1. Introduction	2
1.1. Requirements Language	2

2.	Terminology	2
3.	Border Discovery	3
3.1.	All Ports Discovery	3
3.2.	WAN Port defined As External	4
4.	Home Networking Scenarios	4
4.1.	7084 to Homenet	4
4.1.1.	Addressing	4
4.1.2.	Routing	4
4.1.3.	Border	5
4.1.4.	Service Discovery into the Homenet	5
4.2.	Homenet to 7084	5
4.2.1.	Addressing	6
4.2.2.	Routing	6
4.2.3.	Border	6
4.2.4.	Service Discovery into the Homenet	6
4.3.	Service Provider Edge Router (SPER) to Homenet	7
4.3.1.	Addressing	7
4.3.2.	Routing	7
4.3.3.	Border	7
4.3.4.	Service Discovery	8
4.4.	Homenet to SPER	8
4.4.1.	Addressing	8
4.4.2.	Routing	9
4.4.3.	Border	9
4.4.4.	Service Discovery into the Homenet	9
5.	Security Considerations	9
6.	IANA Considerations	9
7.	Acknowledgements	9
8.	References	9
8.1.	Normative References	9
8.2.	Informative References	11
	Author's Address	11

1. Introduction

This document defines the interactions between the future Homenet network and 7084 Routers and Service Provider Edge Routers (SPER). In the future the SPER will be full Homenet routers but there will be a period of transition. This document specifies how currently deployed SPER will interact with Homenet architecture [I-D.ietf-homenet-arch]. The goal of this document is to make recommendations on issues uncovered to make the devices work with the future Homenet. These recommendations may result in requirements for the Homenet routers.

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 [RFC2119].

2. Terminology

For purposes of this report the Design Team adopts the following terminology.

- o Border: a point, typically resident on a router, between two networks. A basic example is between the main internal homenet and a guest network. This also defines point(s) at which filtering and forwarding policies for different types of traffic may be applied. For the purpose of this document we use the Default Border Definition [I-D.kline-homenet-default-perimeter] to describe how the Border is discovered.
- o SPER: Service Provider Edge Router: A border router intended for home or small-office use that forwards packet explicitly addressed as defined [I-D.grundemann-homenet-hipnet] or [BBF.TR124] connecting the homenet to a service provider network.
- o Homenet: Home network consisting of routers interacting with each other using a dynamic routing protocol for prefix allocation and reachability. Examples include Prefix Assignment [I-D.arkko-homenet-prefix-assignment] and OSPFv3 Auto-Configuration [I-D.ietf-ospf-ospfv3-autoconfig]
- o Homenet Naming and Service Discovery: The Homenet supports the ability for users and devices to be able to discover devices and services available in the Homenet. Currently the mechanism is undefined but methods such as DNSSD [RFC6763], [SSDP], Hybrid model using [I-D.cheshire-dnssd-hybrid] or DNS-Based Service Discovery using OSPFv3 [I-D.stenberg-homenet-dnssdext-hybrid-proxy-ospf] could be used to solve this issue.
- 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 7084: A router intended for home or small-office use that forwards packet explicitly addressed to itself as defined in [RFC7084]

3. Border Discovery

According to [I-D.kline-homenet-default-perimeter] there are 3 types of product interfaces: external, internal, and mixed. Border Discovery is the process of discovering the interface types. Below we describe the the 3 choices.

3.1. All Ports Discovery

Border Discovery must be performed on all interfaces. Legacy Routers that don't support Homenet will not participate in Border Discovery

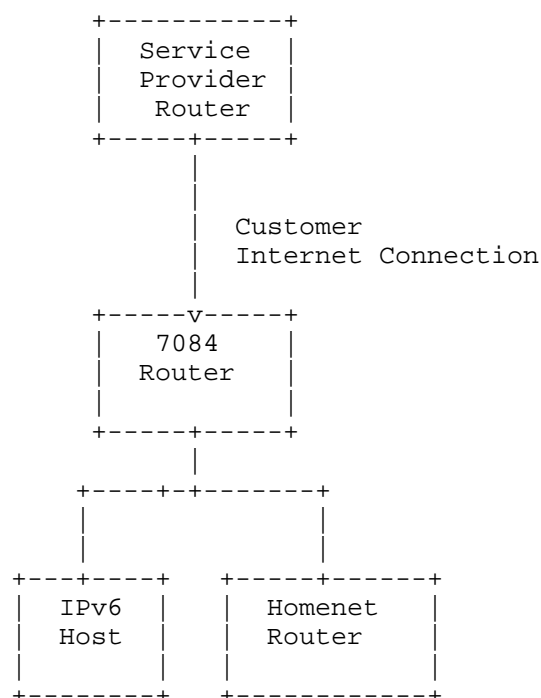
and are considered to be external to the Homenet Border.

3.2. WAN Port defined As External

WAN ports are permanently defined as external requiring no discovery. LAN ports perform Border Discovery. This requires that the user connect the WAN interface to the ISP or SPER defining the boundary. All other ports are in border discovery mode. The advantage of this approach is that it allows the Homenet to have multiple egress ports.

4. Home Networking Scenarios

4.1. 7084 to Homenet



4.1.1. Addressing

A 7084 Router acquires addresses to provision the LAN through DHCP Prefix Delegation [RFC3633]. A 7084 Router will assign a separate /64 from the set of delegated prefix(es) for each LAN interfaces. The Router can assign addresses to the LAN hosts using either SLAAC or DHCP. There is no requirement for redistributing any unused prefix(es) that were delegated to the 7084 Router. Support of IA_PD on the LAN interface is not required for a 7084 Router. If a 7084 Router does not support IA_PD on the LAN interface the Homenet will not receive a prefix allocation, and therefore will not have global addressing for the entire Homenet.

4.1.2. Routing

A 7084 Router learns default routes through Router Advertisements on the WAN interface. Routes are installed when a prefix is assigned to a LAN interface. All other Home Routing information requires user configuration.

A 7084 Router will NOT forward packets from an unrecognized source address. Any IPv6 packets routed from the Homenet would receive an ICMPv6 Destination Unreachable message. This restricts the Homenet to internal communications only. Packets with unrecognized destination addresses in the Homenet MAY pass thru a 7084 Router if configured. This configuration might be done thru the mechanism such as IA_PD or direct configuration.

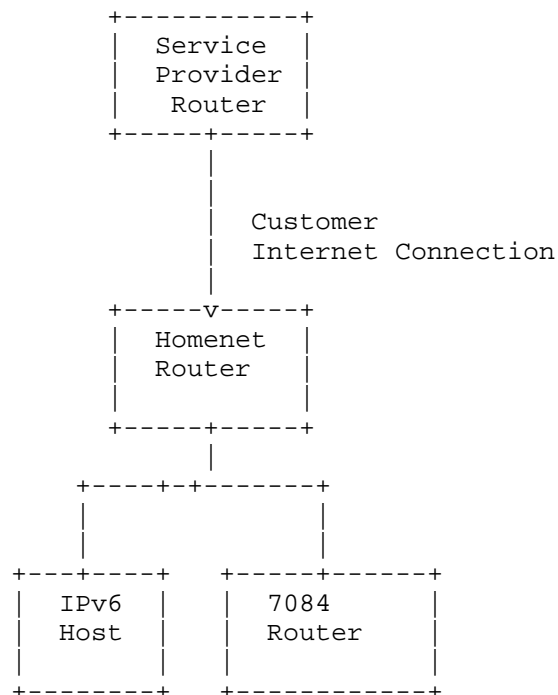
4.1.3. Border

A 7084 Router does not have a method for participating in Homenet border discovery. A 7084 Router and any hosts connected to the Router are considered to be as External to the Homenet. A Homenet Router is recommended to support a configuration method that will allow the border to include the 7084 Router as Internal to the Homenet.

4.1.4. Service Discovery into the Homenet

For service discovery to work routers need to forward multicast traffic appropriately enabling server discovery across the home network. A 7084 Router does not have any requirements for supporting multicast forwarding. Based on this knowledge it is unlikely that Service Discovery between the 7084 and Homemnet will work.

4.2. Homenet to 7084



4.2.1. Addressing

A 7084 Router needs to receive an IA_PD to allow devices on LAN interfaces to be addressed. For addressing to work properly the Homenet must provide IA_PDs when requested.

4.2.2. Routing

When a Homenet Router is assigned an IA_PD it MUST install routes for the prefixes into the Homenet Routing infrastructure. This will allow packets to be routed from the Homenet to the 7084 Router. A 7084 Router only needs a Router Advertisement with a valid Router Lifetime to route into the Homenet.

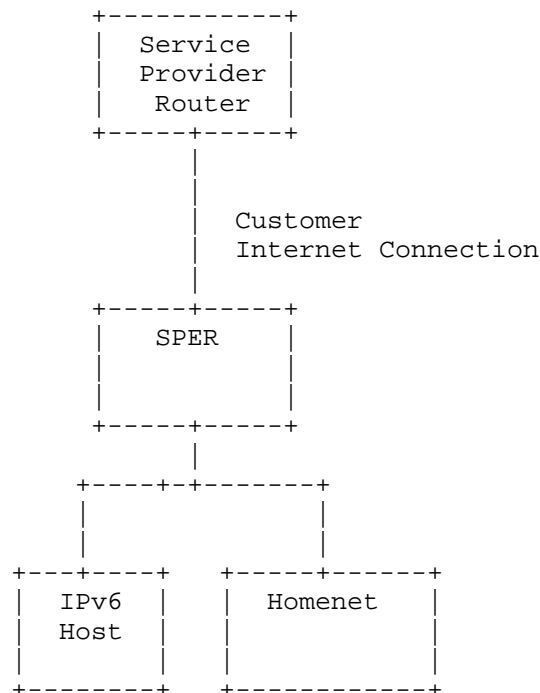
4.2.3. Border

A Homenet Router with the firewall on might not allow valid traffic from devices connected to the 7084 Router. When a Homenet Router is assigned an IA_PD there needs to be a secure way for the Homenet Border to allow IPv6 traffic to flow from the 7084 router into the Homenet or Internet.

4.2.4. Service Discovery into the Homenet

For service discovery to work routers need to forward multicast traffic appropriately enabling server discovery across the home network. A 7084 Router does not have any requirements for supporting multicast forwarding. Based on this knowledge it is unlikely that Service Discovery between the 7084 and Homemnet will work.

4.3. Service Provider Edge Router (SPER) to Homenet



4.3.1. Addressing

SPERs use DHCPv6 prefix sub-delegation to build the network [I-D .grundemann-homenet-hipnet]. If the prefix is larger than a single /64 prefix the SPER will subdivide the IPv6 prefix received via DHCPv6 [RFC3315]. Using Recursive Prefix Delegation allows the Homenet to receive prefixes that can be used to address the network.

4.3.2. Routing

Leveraging the recursive prefix delegation method described above, a SPER installs a route to the WAN interface of the router which delegated the prefixes. With this routing information the SPER is able to properly route packets to and from the Homenet.

4.3.3. Border

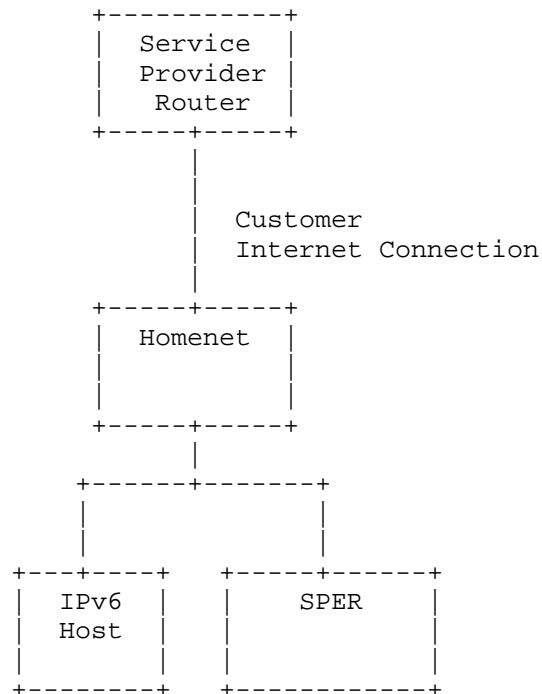
A SPER implements a stateful [RFC6092] firewall which may be have it enabled. This stateful firewall will allow homenet traffic to leave the network. It is limited to only returning traffic originated from the Homenet. No connections can be originated from outside of the Homenet.

A Homenet Router with the firewall on might not allow valid traffic from devices connected to the HIPnet SPER. A Homenet Router will be able to detect a SPER based on a CER_ID, [I-D.donley-dhc-cer-id-option], SPER MUST include an CER_ID option with an address that is not the unspecified address (:::). This allows for the Homenet Router to detect a SPER allowing native IPv6 traffic through the firewall so that traffic can flow between the SPER and Homenet.

4.3.4. Service Discovery

Both the Homenet and SPER have several common protocols that can be used for service discovery such as mDNS [RFC6762], DNS-SD [RFC6763], and [SSDP]. Both the SPER and Homenet Routers may have host directly connected that are using them as DNS servers. If the SPER advertises itself as the DNS-SD server for connected host, the host could query the SPER. The issue that arises with this configuration is the HIPnet Router currently has no method for finding the Homenet router to query when trying to resolve DNS.

4.4. Homenet to SPER



4.4.1. Addressing

A SPER needs to receive an IA_PD to address IPv6 host and routers behind it. If a large enough prefix is assigned, /56 for example, the SPER will attempt further sub-delegation. This will not be optimized for the network but will still function properly. For addressing between the SPER and Homenet to work properly the Homenet must provide IA_PDs when requested.

4.4.2. Routing

When a Homenet Router assigns an IA_PD to the SPER it MUST install routes for the prefixes into the Homenet Routing infrastructure. This will allow packets to be routed from the Homenet to the SPER. If there are two ingress paths to the SPER, the sub-optimal path will be chosen based on the interface that assigned the IA_PD.

4.4.3. Border

A Homenet Router with the firewall enabled might not allow valid traffic from devices connected to the SPER or addressed by the SPER to enter the Homenet. When a Homenet Router assigns an IA_PD there needs to be a secure way for the Homenet Border to allow IPv6 traffic to flow from the SPER into the Homenet or Internet.

4.4.4. Service Discovery into the Homenet

For service discovery to work routers need to forward multicast traffic appropriately enabling server discovery across the home network.

5. Security Considerations

6. IANA Considerations

This document makes no request of IANA.

7. Acknowledgements

The Homenet Design Team: Mikael Abrahamsson, Ray Bellis, John Brzozowski, Lorenzo Colitti, Tim Chown, Chris Donley, Markus Stenberg, Andrew Yourtchecko, Erik Kline

8. References

8.1. Normative References

[I-D.arkko-homenet-prefix-assignment]
Arkko, J., Lindem, A. and B. Paterson, "Prefix Assignment in a Home Network", Internet-Draft draft-arkko-homenet-prefix-assignment-04, May 2013.

[I-D.cheshire-dnssd-hybrid]

Cheshire, S., "Hybrid Unicast/Multicast DNS-Based Service Discovery", Internet-Draft draft-cheshire-dnssd-hybrid-01, January 2014.

[I-D.donley-dhc-cer-id-option]

Donley, C., Klobardans, M., Brzozowski, J. and C. Grundemann, "Customer Edge Router Identification Option", Internet-Draft draft-donley-dhc-cer-id-option-02, January 2014.

[I-D.grundemann-homenet-hipnet]

Grundemann, C., Donley, C., Brzozowski, J., Howard, L. and V. Kuarsingh, "A Near Term Solution for Home IP Networking (HIPnet)", Internet-Draft draft-grundemann-homenet-hipnet-01, February 2013.

[I-D.ietf-homenet-arch]

Chown, T., Arkko, J., Brandt, A., Troan, O. and J. Weil, "IPv6 Home Networking Architecture Principles", Internet-Draft draft-ietf-homenet-arch-11, October 2013.

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

Lindem, A. and J. Arkko, "OSPFv3 Auto-Configuration", Internet-Draft draft-ietf-ospf-ospfv3-autoconfig-05, October 2013.

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

Kline, E., "Default Border Definition", Internet-Draft draft-kline-homenet-default-perimeter-00, March 2013.

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

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

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

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

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

[RFC6092] Woodyatt, J., "Recommended Simple Security Capabilities in Customer Premises Equipment (CPE) for Providing Residential IPv6 Internet Service", RFC 6092, January 2011.

- [RFC6204] Singh, H., Beebee, W., Donley, C., Stark, B. and O. Troan, "Basic Requirements for IPv6 Customer Edge Routers", RFC 6204, April 2011.
- [RFC6763] Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", RFC 6763, February 2013.
- [RFC7084] Singh, H., Beebee, W., Donley, C. and B. Stark, "Basic Requirements for IPv6 Customer Edge Routers", RFC 7084, November 2013.

8.2. Informative References

- [BBF.TR124] Broadband Forum, "TR-124: Functional Requirements for Broadband Residential Gateways Devices", August 2012.
- [RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, February 2013.
- [SSDP] UPnP Forum, "Universal Plug and Play (UPnP) Device Architecture 1.1", November 2008.

Author's Address

Timothy Winters, editor
UNH-IOL
Durham, NH

Email: twinters@iol.unh.edu