Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 6, 2018

T. Lemon
Nibbhaya Consulting
D. Migault
Ericsson
S. Cheshire
Apple Inc.

March 5, 2018

Simple Homenet Naming and Service Discovery Architecture
draft-ietf-homenet-simple-naming-01

Abstract

This document describes how names are published and resolved on homenets, and how hosts are configured to use these names to discover services on homenets. It presents the complete architecture, and describes a simple subset of that architecture that can be used in low-cost 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 https://datatracker.ietf.org/drafts/current/.

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

This Internet-Draft will expire on September 6, 2018.

Table of Contents

1. Introduction

This document is a homenet architecture document. The term 'homenet' refers to a set of technologies that allow home network users to have a local-area network (LAN) with more than one physical link and, optionally, more than one internet service provider. Home network users are assumed not to be knowledgable in network operations, so homenets automatically configure themselves, providing connectivity and service discovery within the home with no operator intervention. This document describes the aspect of homenet automatic configuration that has to do with service discovery and name resolution.

The homenet naming architecture consists of two parts: the simple naming architecture, and the advanced naming architecture. The advanced architecture provides approximate parity of features with a managed network, including the ability to publish services on the internet. The simple architecture provides a minimal set of features required to enable seamless service discovery on a multi-link home

network, but does not attempt to provide feature parity with a managed LAN.

This document begins by presenting a motivational list of requirements and considerations, which should give the reader a clear idea of the scope of the problem being solved. It then explains how each requirement is addressed, and provides references for relevant standards documents describing the details of the implementation. Some requirements are not satisfied by the simple architecture; these are discussed in this document, but explained in more detail in the Advanced Homenet Naming Architecture document, which is to follow.

2. Requirements

Name service on a local area network (LAN) requires the following:

- Name: a forward domain under which information about local services will be published
- Authority: a name server that is authoritative for at least a forward and one or two reverse domains that are applicable to that network
- Resolution: a full-service caching DNS resolver
- Publication: a mechanism that
  - allows services on the LAN to publish information about the services they provide
  - allows services to publish information on how to reach them
  - manages the lifetime of such information, so that it persists long enough to prevent spoofing, but protects end users from seeing stale information
- Host configuration: one or more automatic mechanisms (e.g. DHCP or RA) that provide:
  - caching resolver information to hosts on the LAN
  - information about how services on the LAN can publish information
- Trust: some basis for trusting the information that is provided by the service discovery system

2.1. Managed LAN versus Homenet

On a managed LAN, many of these services can be provided by operators. When a new printer is added to the network, it can be added to the service discovery system (the authoritative server) manually. When a printer is taken out of service, it can be removed. In this scenario, the role of "publisher" is filled by the network operator.

In many managed LANs, establishment of trust for service discovery is simply on the basis of a belief that the local resolver will give a correct answer. Once the service has been discovered and chosen, there may be some security (e.g., TLS) that protects the connection to the service, but the trust model is often just "you're connected to a network you trust, so you can trust the printer that you discovered on this network."

A homenet does not have an operator, so functions that would normally be performed by the operator have to happen automatically. This has implications for trust establishment--since there is no operator controlling what services are published locally, some other mechanism is required for basic trust establishment. Additionally, whereas in a managed LAN with multiple links to the Internet, the network operator can configure the network so that multihoming is handled seamlessly, in a homenet, multihoming must be handled using multiple provisioning domains [RFC7556].

2.2. Homenet-specific considerations

A naming architecture for homenets therefore adds the following considerations:

- All of the operations mentioned here must reliably function automatically, without any user intervention or debugging.

- Because user intervention cannot be required, naming conflicts must be resolved automatically, and, to the extent possible, transparently.

- Devices that provide services must be able to publish those services on the homenet, and those services must be available from any part of the homenet, not just the link to which the device is attached.

- Homenets must address the problem of multiple provisioning domains, in the sense that the DNS may give a different answer depending on whether caching resolvers at one ISP or another are queried.

An additional requirement from the Homenet Architecture [9] is that hosts are not required to implement any homenet-specific capabilities in order to discover and access services on the homenet. This architecture may define optional homenet-specific features, but hosts that do not implement these features must work on homenets.

3. Terminology

This document uses the following terms and abbreviations:

HNR Homenet Router

SHNR Homenet Router implementing simple homenet naming architecture

AHNR Homenet Router implementing advanced homenet naming architecture

ISP Internet Service Provider

4. Name

In order for names to be published on a homenet, it is necessary that there be a set of domain names under which such names are published. These domain names, together, are referred to as the "local domains." By default, homenets use the reserved domain 'home.arpa.' for publishing names for forward lookups. So a host called 'example' that published its name on the homenet would publish its records on the domain name 'example.home.arpa.'. Because 'home.arpa.' is used by all homenets, it has no global meaning, and names published under the domain 'home.arpa' cannot be used outside of the homenet on which they are published.

Homenet routers that implement advanced homenet naming may also be configured with a global domain. How such a domain is configured is out of scope for this document, and is described in the Advanced Homenet Naming Architecture document [advanced].

In addition to the name, which defaults to 'home.arpa.', names are needed for reverse lookups. These names are dependent on the IP addressing used on the homenet. If the homenet is addressed with IPv4, a reverse domain corresponding to the IPv4 subnet [1] section 5.2.1 should be constructed. For example, if the homenet is allocating local IP addresses out of net 10 [3], a domain, '10.in-addr.arpa' would be required. Like 'home.arpa.', '10.in-addr.arpa' is a locally-served zone, and has no validity outside of the homenet.

If the homenet is addressed with IPv6, it is expected to have a unique local address prefix; subsets of this prefix will be

advertised on every link on the homenet. Every service on the homenet that supports IPv6 is expected to be reachable at an address that is configured using the ULA prefix. Therefore there is no need for any IPv6 reverse zone to be populated other than the ULA zone. So for example if the homenet's ULA prefix is fd00:2001:db8::/48, then the reverse domain name for the homenet would end in '8.b.d.0.1.0.0.2.0.0.d.f.ip6.arpa'.

5. Authority

The authority role is provided by a name server that is authoritative for each of the local domains. SHNRs provide authoritative service for the homenet using DNSSD Discovery Broker [17]. SHNRs also provide Discovery Relay service [12]. On a homenet that has only SHNRs, each SHNR individually provides authoritative service for the whole homenet by using Discovery relays to discover services off the local link.

The Discovery Proxy model relies on each link having its own name. However, homenets do not actually have a way to name local links that will make any sense to the end user. Consequently, this mechanism will not work without some tweaks. In order to address this, homenets will use Discovery Brokers [17]. The discovery broker will be configured so that a single query for a particular service will be successful in providing the information required to access that service, regardless of the link it is on.

Artificial link names will be generated using HNCP. These should only be visible to the user in graphical user interfaces in the event that the same name is claimed by a service on two links. Services that are expected to be accessed by users who type in names should use [13] if it is available.

It is possible that local services may offer services available on IP addresses in public as well as ULA prefixes. Homenet hybrid proxies MUST filter out global IP addresses, providing only ULA addresses, similar to the process described in section 5.5.2 of [11].

This filtering applies to queries within the homenet; it is appropriate for non-ULA addresses to be used for offering services, because in some cases end users may want such services to be reachable outside of the homenet. Configuring this is however out of scope for this document.

6. Resolution

Name resolution is provided by a local DNS cache or proxy on the homenet, henceforth the "local resolver." All host queries are sent to this local resolver. The local resolver may either act as a full-service caching resolver, or as a DNS proxy. Its responsibility with respect to queries on the homenet is to notice queries for names for which the local authoritative server is authoritative. Queries for such names are handled through the local authoritative server. Queries for all other names are resolved either by forwarding them to an ISP-provided full service resolver, or by providing the full service resolver function locally.

7. Publication

7.1. DNS Service Discovery Registration Protocol

The DNSSD Service Registration protocol [13] requires that DNS updates be validated on the basis that they are received on the local link. To ensure that such registrations are actually received on local links in the homenet, updates are sent to the local relay proxy ([12]) (XXX how?).

The relay proxy encapsulates the update and sends it to whatever Discovery Proxy is listening on the link; the Discovery proxy then either consumes the update directly, or forwards it to the authoritative resolver for the local service discovery zone. If the registration protocol is not supported on the homenet, the Discovery Proxy rejects the update with a ??? RCODE.

Homenets are not required to support Service Registration. Service registration requires a stateful authoritative DNS server; this may be beyond the capability of the minimal Homenet router. However, more capable Homenet routers should provide this capability. In order to make this work, minimal Homenet routers MUST implement the split hybrid proxy [12]. This enables a Homenet with one or more Homenet routers that provide a stateful registration cache to allow those routers to take over service, using Discovery Relays to service links that are connected using Homenet routers with more limited functionality.

7.2. Configuring Service Discovery

Clients discovering services using DNS-SD [7] follow a two-step process. The first step is for the client device to determine in which domain(s) to attempt to discover services. The second step is for the client device to then seek desired service(s) in those domain(s). For an example of the second step, given the desired

service type "IPP Printing", and the domains "local" and "meeting.ietf.org", the client device forms the queries "_ipp._tcp.local. PTR ?" (resolved using Multicast DNS) and "_ipp._tcp.meeting.ietf.org PTR. ?" (resolved using Unicast DNS) and then presents the combined list of results to the user.

The first step, determining in which domain(s) to attempt to discover services, is performed in a variety of ways, as described in Section 11 of the DNS-Based Service Discovery specification [7].

The domain "local" is generally always in the set of domains in which the client devices attempt to discover services, and other domains for service discovery may be configured manually by the user.

The device also learns additional domains automatically from its network environment. For this automatic configuration discovery, special DNS queries are formulated. To learn additional domain(s) in which to attempt to discover services, the query string "lb._dns_sd._udp" is prepended onto three different kinds of "bootstrap domain" to form DNS queries that allow the device to learn the configuration information.

One of these bootstrap domains is the fixed string "local". The device issues the query "lb._dns_sd._udp.local. PTR ?" (resolved using Multicast DNS), and if any answers are received, then they are added to the set of domains in which the client devices attempt to discover services.

Another kind of these bootstrap domains is name-based, derived from the DHCPv4 "domain name" option (code 15) [4] (for IPv4) or the DNS Search List (DNSSL) Router Advertisement option [10] (for IPv6). If a domain in the DNSSL is "example.com", then the device issues the query "lb._dns_sd._udp.example.com. PTR ?" (resolved using Unicast DNS), and if any answers are received, then they are likewise added to the set of domains in which the client devices attempt to discover services.

Finally, the third kind of bootstrap domain is address-based, derived from the device's IP address(es) themselves. If the device has IP address 192.168.1.100/24, then the device issues the query "lb._dns_sd._udp.0.1.168.192.in-addr.arpa. PTR ?" (resolved using Unicast DNS), and if any answers are received, then they are also added to the set of domains in which the client devices attempt to discover services.

Since there is an HNR on every link of a homenet, automatic configuration could be performed by having HNRs answer the "lb._dns_sd._udp.local. PTR ?" (Multicast DNS) queries. However,

because multicast is slow and unreliable on many modern network technologies like Wi-Fi, we prefer to avoid using it. Instead we require that a homenet be configured to answer the name-based bootstrap queries. By default the domain in the DNSSL communicated to the client devices will be "home.arpa", and the homenet will be configured to correctly answer queries such as "lb._dns_sd._udp.example.com. PTR ?", though client devices must not assume that the name will always be "home.arpa". A client could be configured with any valid DNSSL, and should construct the appropriate bootstrap queries derived from the name(s) in their configured DNS Search List.

HNRs will answer domain enumeration queries against every IPv4 address prefix advertised on a homenet link, and every IPv6 address prefix advertised on a homenet link, including prefixes derived from the homenet's ULA(s). Whenever the "<domain>" sequence appears in this section, it references each of the domains mentioned in this paragraph.

Homenets advertise the availability of several browsing zones in the "b._dns_sd._udp.<domain>" subdomain. By default, the 'home.arpa' domain is advertised. Similarly, 'home.arpa' is advertised as the default browsing and service registration domain under "db._dns_sd._udp.<domain>", "r._dns_sd._udp.<domain>", "dr._dns_sd._udp.<domain>" and "lb._dns_sd._udp.<domain>".

In order for this discovery process to work, the homenet must provide authoritative answers for each of the domains that might be queried. To do this, it provides authoritative name service for the 'ip6.arpa' and 'in-addr.arpa' subdomains corresponding to each of the prefixes advertised on the homenet. For example, consider a homenet with the 192.168.1.0/24, 2001:db8:1234:5600::/56 and fc01:2345:6789:1000::/56 prefixes. This homenet will have to provide a name server that claims to be authoritative for 1.168.192.in-addr.arpa, 6.5.4.3.2.1.8.b.d.0.1.0.0.2.ip6.arpa and 0.0.9.8.7.6.5.4.3.2.1.0.c.f.ip6.arpa.

An IPv6-only homenet would not have an authoritative server for a subdomain of in-addr.arpa. These public authoritative zones are required for the public prefixes even if the prefixes are not delegated. However, they need not be accessible outside of the homenet.

It is out of the scope of this document to specify ISP behavior, but we note that ISPs have the option of securely delegating the zone, or providing an unsigned delegation, or providing no delegation. Any delegation tree that does not include an unsigned delegation at or

above the zone cut for the ip6.arpa reverse zone for the assigned prefix will fail to validate.

Ideally, an ISP should provide a secure delegation using a zone-signing key provided by the homenet. However, that too is out of scope for this document. Therefore, an ISP that wishes to support users of the simple homenet naming architecture will have to provide an unsigned delegation. We do not wish, however, to discourage provisioning of signed delegations when that is possible.

8. Host Configurtion

Hosts on the homenet receive a set of resolver IP addresses using either DHCP or RA. IPv4-only hosts will receive IPv4 addresses of resolvers, if available, over DHCP. IPv6-only hosts will receive resolver IPv6 addresses using either stateful (if available) or stateless DHCPv6, or through the Recursive DNS Server Option ([10], Section 5.1) in router advertisements.

All Homenet routers provide resolver information using both stateless DHCPv6 and RA; support for stateful DHCPv6 and DHCPv4 is optional, however if either service is offered, resolver addresses will be provided using that mechanism as well.

9. Globally Unique Name

Automatic configuration of a globally unique name for the homenet is out of scope for this document. However, homenet servers MUST allow the user to configure a globally unique name in place of the default name, 'home.arpa.' By default, even if configured with a global name, homenet routers MUST NOT answer queries from outside of the homenet for subdomains of that name.

10. DNSSEC Validation

DNSSEC Validation for the 'home.arpa' zone and for the locally-served 'ip6.arpa and 'in-adr.arpa' domains is not possible without a trust anchor. Establishment of a trust anchor for such validation is out of scope for this document.

Homenets that have been configured with a globally unique domain MUST support DNSSEC signing of local names, and must provide a way to generate a KSK that can be used in the secure delegation of the globally unique domain assigned to the homenet.

11. Support for Multiple Provisioning Domains

Homenets must support the Multiple Provisioning Domain Architecture [9]. Hosts connected to the homenet may or may not support multiple provisioning domains. For hosts that do not support multiple provisioning domains, the homenet provides one or more resolvers that will answer queries for any provisioning domain. Such hosts may receive answers to queries that either do not work as well if the host chooses a source address from a different provisioning domain, or does not work at all. However, the default source address selection policy, longest-match [CITE], will result in the correct source address being chosen as long as the destination address has a close match to the prefix assigned by the ISP.

Hosts that support multiple provisioning domains will be provisioned with one or more resolvers per provisioning domain. Such hosts can use the IP address of the resolver to determine which provisioning domain is applicable for a particular answer.

Each ISP has its own provisioning domain. Because ISPs connections cannot be assumed to be persistent, the homenet has its own separate provisioning domain.

Configuration from the IPv4 DHCP server are treated as being part of the homenet provisioning domain. The case where a homenet advertises IPv4 addresses from one or more public prefixes is out of scope for this document. Such a configuration is NOT RECOMMENDED for homenets.

Configuration for IPv6 provisioning domains is done using the Multiple Provisioning Domain RA option [CITE].

12. Using the Local Namespace While Away From Home

This architecture does not provide a way for service discovery to be performed on the homenet by devices that are not directly connected to a link that is part of the homenet.

13. Management Considerations

This architecture is intended to be self-healing, and should not require management. That said, a great deal of debugging and management can be done simply using the DNS Service Discovery protocol.

14. Privacy Considerations

Privacy is somewhat protected in the sense that names published on the homenet are only visible to devices connected to the homenet. This may be insufficient privacy in some cases.

The privacy of host information on the homenet is left to hosts. Various mechanisms are available to hosts to ensure that tracking does not occur if it is not desired. However, devices that need to have special permission to manage the homenet will inevitably reveal something about themselves when doing so. It may be possible to use something like HTTP token binding [15] to mitigate this risk.

15. Security Considerations

There are some clear issues with the security model described in this document, which will be documented in a future version of this section. A full analysis of the avenues of attack for the security model presented here have not yet been done, and must be done before the document is published.

16. IANA considerations

No new actions are required by IANA for this document.

Note however that this document is relying on the allocation of 'home.arpa' described in Special Use Top Level Domain '.home.arpa' [16]. This document therefore can't proceed until that allocation is done. [RFC EDITOR PLEASE REMOVE THIS PARAGRAPH PRIOR TO PUBLICATION].

17. Normative References

[1] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <https://www.rfc-editor.org/info/rfc1034>.

[2] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <https://www.rfc-editor.org/info/rfc1035>.

[3] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, DOI 10.17487/RFC1918, February 1996, <https://www.rfc-editor.org/info/rfc1918>.

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

[5] Vixie, P., Ed., Thomson, S., Rekhter, Y., and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)", RFC 2136, DOI 10.17487/RFC2136, April 1997, <https://www.rfc-editor.org/info/rfc2136>.

[6] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, DOI 10.17487/RFC6762, February 2013, <https://www.rfc-editor.org/info/rfc6762>.

[7] Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", RFC 6763, DOI 10.17487/RFC6763, February 2013, <https://www.rfc-editor.org/info/rfc6763>.

[8] Chown, T., Ed., Arkko, J., Brandt, A., Troan, O., and J. Weil, "IPv6 Home Networking Architecture Principles", RFC 7368, DOI 10.17487/RFC7368, October 2014, <https://www.rfc-editor.org/info/rfc7368>.

[9] Anipko, D., Ed., "Multiple Provisioning Domain Architecture", RFC 7556, DOI 10.17487/RFC7556, June 2015, <https://www.rfc-editor.org/info/rfc7556>.

[10] Jeong, J., Park, S., Beloeil, L., and S. Madanapalli, "IPv6 Router Advertisement Options for DNS Configuration", RFC 8106, DOI 10.17487/RFC8106, March 2017, <https://www.rfc-editor.org/info/rfc8106>.

[11] Cheshire, S., "Discovery Proxy for Multicast DNS-Based Service Discovery", draft-ietf-dnssd-hybrid-07 (work in progress), September 2017.

[12] Cheshire, S. and T. Lemon, "Multicast DNS Discovery Relay", draft-sctl-dnssd-mdns-relay-02 (work in progress), November 2017.

[13] Cheshire, S. and T. Lemon, "Service Registration Protocol for DNS-Based Service Discovery", draft-sctl-service-registration-00 (work in progress), July 2017.

[14] Korhonen, J., Krishnan, S., and S. Gundavelli, "Support for multiple provisioning domains in IPv6 Neighbor Discovery Protocol", draft-ietf-mif-mpvd-ndp-support-03 (work in progress), February 2016.

[15] Popov, A., Nystrom, M., Balfanz, D., Langley, A., Harper, N., and J. Hodges, "Token Binding over HTTP", draft-ietf-tokbind-https-12 (work in progress), January 2018.

[16] Pfister, P. and T. Lemon, "Special Use Domain 'home.arpa.'", draft-ietf-homenet-dot-14 (work in progress), September 2017.

[17] Cheshire, S. and T. Lemon, "Service Discovery Broker", draft-sctl-discovery-broker-00 (work in progress), July 2017.

Appendix A. Existing solutions

Previous attempts to automate naming and service discovery in the context of a home network are able to function with varying degrees of success depending on the topology of the home network. Unfortunately, these solutions do not fully address the requirements of homenets.

For example, Multicast DNS [6] can provide naming and service discovery [7], but only within a single multicast domain.

The Domain Name System provides a hierarchical namespace [1], a mechanism for querying name servers to resolve names [2], a mechanism for updating namespaces by adding and removing names [5], and a mechanism for discovering services [7]. Unfortunately, DNS provides no mechanism for automatically provisioning new namespaces, and secure updates to namespaces require that the host submitting the update have a public or symmetric key that is known to the network and authorized for updates. In an unmanaged network, the publication of and authorization of these keys is an unsolved problem.

Some managed networks get around this problem by having the DHCP server do DNS updates. However, this doesn't really work, because DHCP doesn't provide a mechanism for updating service discovery records: it only supports publishing A and AAAA records.

This partially solves the trust problem: DHCP can validate that a device is at least connected to a network link that is actually part of the managed network. This prevents an off-network attacker from registering a name, but provides no mechanism for actually validating the identity of the host registering the name. For example, it would be easy for an attacker on the network to steal a registered name.

Hybrid Multicast DNS [11] proposes a mechanism for extending multicast DNS beyond a single multicast domain. However, in order to use this as a solution, some shortcomings need to be considered.

Most obviously, it requires that every multicast domain have a separate name. This then requires that the homenet generate names for every multicast domain. These names would then be revealed to the end user. But since they would be generated automatically and arbitrarily, they would likely cause confusion rather than clarity, and in degenerate cases requires that the end user have a mental model of the topology of the network in order to guess on which link a given service may appear.

At present, the approach we intend to take with respect to disambiguation is that this will not be solved at a protocol level for devices that do not implement the registration protocol.

Authors' Addresses

Ted Lemon
Nibbhaya Consulting
P.O. Box 958
Brattleboro, Vermont 05301
United States of America

Email: mellon@fugue.com

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

Email: daniel.migault@ericsson.com

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

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