

Internet Engineering Task Force  
Internet-Draft  
Intended status: Informational  
Expires: May 12, 2019

S. Cheshire  
Apple Inc.  
T. Lemon  
Nibbhaya Consulting  
November 8, 2018

Unicast Service Discovery Autoconfiguration  
draft-sctl-dnssd-unicast-autoconfig-00

## Abstract

This document considers the requirements for adding a Thread mesh to an existing home network, where the infrastructure of that existing home network was designed with no prior knowledge of Thread, and provides no special or unusual facilities designed to support this.

## 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 May 12, 2019.

## Copyright Notice

Copyright (c) 2018 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 (<https://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.

---

Internet-Draft      Unicast Service Discovery Autoconfig      November 2018

## 1. Introduction

[Authors' note: As an initial draft, in places this document presents several alternatives that are being considered. We invite feedback and comments to help evolve this document.]

Because multicast can be inefficient and unreliable [[Mcast](#)], work is taking place to enable DNS-Based Service Discovery [[RFC6763](#)] to operate with less reliance on multicast [[Roadmap](#)]. One current target use case for this work is Thread [[Thread](#)] wireless mesh networking.

Thread wireless mesh networking uses IEEE 802.15.4 radios, which use little power, and are suitable for battery-powered devices. The Thread protocol organizes the network nodes into a mesh, typically with a Thread border router that connects the mesh to the home network. For the purposes of this document we will refer to the home network, be it Ethernet, or Wi-Fi, or both, or other similar technologies, simply as the home network. The home network forms a backbone to which one or more Thread mesh networks connect via Thread border routers.

Existing work describes how DNS-Based Service Discovery can be performed using unicast on such a network. Devices on the Thread mesh offering services use Service Registration Protocol [[RegProt](#)] to register their services at a Service Registration Server. Devices seeking to discover these services send unicast queries to the Service Registration Server using unicast DNS [[RFC1034](#)] [[RFC1035](#)] for single individual queries, and using DNS Push Notifications [[Push](#)] where ongoing change notification is required.

Certain configuration information is required for this to work. Devices on the Thread mesh offering services need to know what names to use when registering those services, and to what address they should send their service registrations. Devices seeking to discover these services need to know what names to use when constructing their queries, and to what address they should send those queries. In addition, IPv6 address prefixes need to be chosen and configured for both the home network and the Thread mesh network(s), and communicated, in order to facilitate unicast communication between clients and the services they have discovered.

For proof-of-concept experiments, the necessary information can be

configured manually, and this has been done successfully. For deployment, we need to determine how the necessary information will be learned and configured automatically in real-world scenarios.

The Thread wireless mesh protocol includes mechanisms to perform configuration tasks on the mesh, like electing a lead router, and communicating this information to devices on the Thread mesh. This existing mechanism can be extended fairly simply to facilitate the necessary Service Registration Protocol configuration tasks. The Service Registration Protocol [[RegProt](#)] specification document advocates that if a device offering a service has no information regarding the domain in which to register that service, it should use the special use domain name [[RFC6761](#)] "services.arpa" to indicate that the Service Registration Server should substitute a domain of its choice, and that same mechanism is recommended in this case.

On the home network side of the Thread border router, there are several possibilities. The necessary configuration tasks could be handled by the home network's main gateway, by a collection of Homenet routers using HNCP, or independently by the Thread border router.

### [1.1](#). Configuration by Main Gateway

The home network's main gateway could handle the necessary configuration tasks.

The main gateway could be responsible for selecting IPv6 address prefixes for each of the links in the network, and communicating that information to the relevant routers, perhaps using DHCPv6 prefix delegation.

The information about what domain name to use for service discovery can be communicated to client devices on the home network using DHCP or IPv6 router advertisement options. Currently this is done using the respective "DNS search list" options, though new options for this specific purpose could be defined in the future. If the user has a registered globally unique domain name for this purpose and the main gateway is configured with this information, then that domain name can be communicated to client devices. In the absence of a

registered globally unique domain name the special-use domain name [[RFC6761](#)] "home.arpa" [[RFC8375](#)] should be used as a reasonable out-of-the-box default.

Similarly, the information about what DNS recursive resolver to use can be communicated to client devices on the home network using DHCP or IPv6 router advertisement options. If the main gateway configures its own address as the DNS recursive resolver for clients to use, it can ensure that operations using "home.arpa" are handled appropriately. Sending queries for names within "home.arpa" to public recursive resolvers on the Internet will not yield useful results, because names within "home.arpa" are not globally unique.

They are unique only within the local network, and hence queries for those names need to be handled within the local network.

## [1.2.](#) Configuration using HNCP

A complex home network with multiple links and multiple routers could be managed using HNCP. However, at this time, this remains a future possibility, since it is likely to be some time before HNCP is widely used.

## [1.3.](#) Self-Configuration by Thread Border Routers

The previous two scenarios assume that the home network's main gateway, or its HNCP mesh, has specific capabilities to configure and support the use of unicast DNS service discovery.

An alternative scenario is to consider the case where a Thread border router is added to an existing home network, which has no special mechanisms in place to support this operation.

The remainder of this document explores this scenario.

One possibility to keep in mind is that in this scenario, adding one or more Thread border routers to an existing home network that doesn't itself use HNCP, the Thread border router(s) themselves could use HNCP as the protocol to communicate between each other to coordinate their operation on the network.

## [2.](#) Adding Thread Mesh to Existing Home Network

This section explores the requirements for connecting a Thread mesh, via a Thread border router, to a typical home network. For the purposes of this document, it is assumed that the existing network infrastructure is fixed and cannot be changed. Changes or new functionality may be implemented as required in the Thread devices on the Thread mesh, in the Thread border router, or in the devices on the home network that will be communicating with the Thread devices. Since this document assumes no changes to the existing network infrastructure, it is necessary to state the assumptions about that existing network infrastructure.

We consider a typical home network to be a single multicast/broadcast domain. If there are multiple Ethernet switches or Wi-Fi access points, they are configured so that together they provide a single logical link. If there is a NAT gateway, it is at the network egress point. (A NAT gateway on the path between two devices on a home network makes communication between those two devices considerably more complicated, and this document does not address that case.)

In order to add a Thread mesh usefully to an existing home network,

several things need to be accomplished. The goal is to accomplish these objectives without requiring changes to the existing infrastructure on that home network.

1. Delivery of unicast traffic in both directions, from home network to Thread mesh, and from Thread mesh to home network.
2. Enabling services offered by devices on the Thread mesh to be discovered by clients seeking those services.
3. Enabling services offered by devices on the home network to be discovered by clients on the Thread mesh seeking those services.

### [2.1.](#) Unicast Delivery

If HNCP were in use on the network, then Thread border routers could participate and use HNCP to manage their configuration.

In the absence of HNCP, Thread border routers need a way to self-configure, without assistance from the home network's existing infrastructure.

What is proposed is that Thread border routers select a 32-bit random number, and use that to construct an IPv6 ULA prefix for their connected mesh, which is very likely to be unique in that home. The Thread border router then advertises reachability to that IPv6 ULA prefix onto the home network using IPv6 Router Advertisements. In principle, this can be done independently of whatever other IPv6

prefixes, if any, are being advertised on the home network by the home network's existing main gateway. [It has been reported, however, that there are at least some client devices that do not properly handle receiving multiple independent IPv6 Router Advertisements like this, so some investigation and bug fixing may be required to make this work.]

In the case where there are multiple independent Thread border routers connected to the home network, serving separate Thread meshes, we want to avoid the situation where two different Thread border routers choose the same randomly-selected IPv6 ULA prefix. This can be achieved by having the Thread border routers listen for IPv6 Router Advertisements before selecting their IPv6 ULA prefix. If a Thread border router receives IPv6 Router Advertisements offering reachability to its IPv6 ULA prefix via a different path, then this indicates that an inadvertent duplication may have occurred, and the Thread border router should select a different IPv6 ULA prefix for its mesh.

## [2.2.](#) Discovery of Services on the Thread Mesh

To facilitate unicast discovery of services on the Thread mesh, four things need to be determined:

1. How a device on the Thread mesh, offering services, knows what parent domain name to use when registering services.

2. How that device knows to what address its service registrations should be sent (if the name does not fall under a registered globally unique domain name).
3. How a client device, on the Thread mesh or the home network, seeking services, knows what parent domain name to use querying to discover services.
4. How that device knows to what address its unicast service discovery queries should be sent (if the name does not fall under a registered globally unique domain name).

Devices on the Thread mesh should register services using the parent domain "services.arpa". This indicates that the Service Registration Server should automatically substitute an appropriate domain.

The Thread mesh management protocol can be used to configure devices on the Thread mesh with the address to which they should send their service registrations.

The Thread border router needs to communicate, to devices on the home network, how they can discover services on the Thread mesh.

This involves communicating the service discovery domain. In principle, this could be a registered globally unique domain name, in which case the normal DNS delegation mechanism using NS records allows the client to discover what server is authoritative for those names. In many cases though, the Thread border router will not have a registered globally unique domain name allocated. To provide out-of-the-box automatic operation, the Thread border router needs to be able to generate its own locally unique name to use. The special use domain name "local" is not suitable, because of its implied semantics that these names are resolved using link-local multicast DNS [[RFC6762](#)]. The special use domain name "home.arpa" is not suitable, because of its implied coordination via HNCP, and the home network's main gateway may not support HNCP [[RFC8375](#)]. To provide out-of-the-box automatic operation, this document proposes a new special use domain name "adhoc.arpa" for this purpose. By default a Thread border router will use the name "thread.adhoc.arpa". If this name is

already in use on the home network, then a new unique name will be



selected, such as "thread-2.adhoc.arpa".

The Thread border router needs to communicate the service discovery domain to peers on the home network. In the case that the service discovery domain falls under the "adhoc.arpa" name, the Thread border router also needs to communicate that queries for these names need to be sent to the Thread border router directly, not to the client's default DNS recursive resolver.

Three alternatives are being considered

1. Use link-local Multicast DNS queries and records to convey the service discovery domain, and optionally the address to which queries should be sent.
2. Define a new IPv6 router advertisement option to communicate the service discovery domain, and optionally the address to which queries should be sent.
3. Add this information to the Multiple Provisioning Domain Router Advertisement option [[RFC7556](#)] [[MPvD](#)].

One question to answer is whether the Multicast DNS records or IPv6 router advertisement options would directly convey the domain name to use for service discovery, or a base name used to derive domain enumeration queries of the form lb.\_dns-sd.\_udp.<domain> [[RFC6763](#)].

Another question is whether to use a single Multicast DNS record or IPv6 router advertisement option that communicates both the domain name and the address to use for queries, or a pair of records/options, one carrying the name to use for service discovery, and a second, if necessary, associating an "adhoc.arpa" name with the address to use for those queries.

With the appropriate configuration methods defined, and implemented on client devices, client devices on the home network would discover additional domains to use for service discovery, and send appropriate service discovery queries to Thread border routers on the home network.

The same discovery domain, and optionally the address to which queries should be sent, is communicated to client devices on the Thread mesh using the Thread mesh management protocol.

### [2.3.](#) Discovery of Services on the Home Network

To facilitate devices on the Thread mesh discovering services offered on the home network, advertised using Multicast DNS, a Discovery Proxy [[DisProx](#)] is implemented in the Thread border router.

As above in [Section 2.2](#) the Thread mesh management protocol is used to communicate a discovery domain, and the address to which queries should be sent for that discovery domain, to client devices on the Thread mesh.

The address in this case is the address of the Thread border router. The discovery domain could be some generated unique name under "adhoc.arpa", or it could be some fixed special use domain name. The fixed name could be a simple fixed string like "lan.arpa", or it could be a special reserved name under "adhoc.arpa" such as "services.adhoc.arpa". The latter is probably preferred because it avoids having to request multiple special use domain names [[RFC6761](#)]. Alternatively, we could organize all the required special names such that they fall under a single reserved special use domain name "services.arpa."

When the Thread border router receives a query for a name under this discovery domain, it uses the Discovery Proxy mechanism [[DisProx](#)] to perform Multicast DNS queries on behalf of the client, returning the results to the client.

---

Internet-Draft      Unicast Service Discovery Autoconfig      November 2018

### [3.](#) Security Considerations

As an informational document, this document introduces no new Security Considerations of its own. The various referenced documents each describe their own relevant Security Considerations as appropriate.

### [4.](#) Domain Name Reservation Considerations

As currently envisaged, this document may end up requesting a special use domain name [[RFC6761](#)]. If so, once the special properties are fully determined, this section will be populated with the appropriate text.

### [5.](#) Informative References

- [DisProx] Cheshire, S., "Discovery Proxy for Multicast DNS-Based Service Discovery", [draft-ietf-dnssd-hybrid-08](#) (work in progress), March 2018.
- [Mcast] Perkins, C., McBride, M., Stanley, D., Kumari, W., and J. Zuniga, "Multicast Considerations over IEEE 802 Wireless Media", [draft-ietf-mboned-ieee802-mcast-problems-03](#) (work in progress), October 2018.
- [MPvD] 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.
- [Push] Pusateri, T. and S. Cheshire, "DNS Push Notifications", [draft-ietf-dnssd-push-14](#) (work in progress), March 2018.
- [RegProt] Cheshire, S. and T. Lemon, "Service Registration Protocol for DNS-Based Service Discovery", [draft-sctl-service-registration-00](#) (work in progress), July 2017.
- [RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, [RFC 1034](#), DOI 10.17487/RFC1034, November 1987,

<<https://www.rfc-editor.org/info/rfc1034>>.

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

[RFC6761] Cheshire, S. and M. Krochmal, "Special-Use Domain Names", [RFC 6761](#), DOI 10.17487/RFC6761, February 2013, <<https://www.rfc-editor.org/info/rfc6761>>.

Cheshire & Lemon

Expires May 12, 2019

[Page 10]

---

Internet-Draft      Unicast Service Discovery Autoconfig      November 2018

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

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

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

[RFC8375] Pfister, P. and T. Lemon, "Special-Use Domain 'home.arpa.'", [RFC 8375](#), DOI 10.17487/RFC8375, May 2018, <<https://www.rfc-editor.org/info/rfc8375>>.

[Roadmap] Cheshire, S., "Service Discovery Road Map", [draft-cheshire-dnssd-roadmap-03](#) (work in progress), October 2018.

[Thread] "Thread Wireless Mesh Networking", <<https://www.threadgroup.org/>>.

#### Authors' Addresses

Stuart Cheshire  
Apple Inc.  
One Apple Park Way  
Cupertino, California 95014  
USA

Phone: +1 (408) 996-1010

Email: cheshire@apple.com

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

Email: mellon@fugue.com