Internet Engineering Task Force
Internet-Draft
Intended status: Best Current Practice
Expires: 27 October 2022

T. Lemon
Apple, Inc.

25 April 2022

Self-configuring Stub Networks: Problem Statement
draft-lemon-stub-networks-ps-02

Abstract

IETF currently provides protocols for automatically connecting single hosts to existing network infrastructure. This document describes a related problem: the problem of connecting a stub network (a collection of hosts behind a router) automatically to existing network infrastructure in the same manner.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This document describes the problem of linking stub networks to existing networks automatically, in the same way that hosts, when connected to an existing network, are able to discover network addressing parameters, information about routing, and services that are advertised on the network.

There are several use cases for stub networks. Motivating factors include:

* Transitory connectivity: a mobile device acting as a router for a set of co-located devices could connect to a network and gain access to services for itself and for the co-located devices. Such a stub network is unlikely to have more than one stub router.

* Incompatible media: for example, a constrained 802.15.4 network connected as a stub network to a WiFi or ethernet infrastructure network. In the case of an 802.15.4 network, it is quite possible that the devices used to link the infrastructure network to the stub network will not be conceived of by the end user as routers. Consequently, we cannot assume that these devices will be on all the time. A solution for this use case will require some sort of commissioning process for stub routers, and can’t assume that any particular stub router will always be available; rather, any stub router that is available must be able to adapt to current conditions to provide reachability.

* Convenience: end users often connect devices to each other in order to extend networks

What makes stub networks a distinct type of network is simply that a stub network never provides transit between networks to which it is connected. The term "stub" refers to the way the network is seen by the link to which it is connected: there is reachability through a stub network router to the stub network from that link, but there is no reachability to any link beyond that one.

Stub networks may be globally reachable, or may be only locally reachable. A host on a globally reachable stub network can interoperate with other hosts anywhere on the Internet. A host on a locally reachable stub network can only interoperate with hosts on the network link(s) to which it is connected.

The goal of this document is to describe the minimal set of changes or behaviors required to use existing IETF specifications to support the stub network use case. The result should be a small set of protocol enhancements (ideally no changes at all to protocols) and should be deployable on existing networks without requiring changes to those networks. Both the locally-reachable and globally-reachable use case should be able to be made to work, and ideally the globally-reachable use case should build on what is used to make the locally-reachable use case work, rather than requiring two separate solutions.

1.1. Interoperability Goals

What we mean by "interoperate" is that a host on a stub network:

* is discoverable by applicable hosts that are not on the stub network

* is able to acquire an IP address that can be used to communicate with applicable hosts not on the stub network

* has reachability to the network(s) to which applicable hosts are attached

* is reachable from the network(s) to which applicable hosts are attached

Discoverability here means "discoverable using DNS, or DNS Service Discovery". As an example, when one host connected to a specific WiFi network wishes to discover services on hosts connected to that same WiFi network, it can do so using multicast DNS (RFC6762), which is an example of DNS Service Discovery. Similarly, when a host on some other network wishes to discover the same service, it must use DNS-based DNS Service Discovery [RFC6763]. In both cases, "discoverable using DNS" means that the host has an entry in the DNS.

NOTE: it may be tempting to ask, why do we lump discoverability in with reachability and addressability, both of which are essentially Layer 3 issues? The answer is that it does us no good to automatically set up connectivity between stub network hosts and infrastructure hosts if the infrastructure hosts have no mechanism for learning about the availability of services provided by stub network hosts. For stub networks that only consume cloud services this will not be an issue, but for stub networks that provide services, e.g. the incompatible media use case mentioned earlier, discoverability is necessary in order for stub network connectivity to be useful.

Ability to acquire an IP address that can be used to communicate means that the IP address a host on the stub network acquires can be used to communicate with it by hosts on neighbor networks, for locally reachable stub networks, or by hosts on any network, for globally reachable networks. Various means of providing such addresses are discussed later.

Reachability to networks on which applicable hosts are attached means that when a host on the stub network has the IP address of an applicable host with which it intends to communicate, that host knows of a next-hop router to which it can send datagrams, so that they will ultimately reach the host with that IP address.

Reachability from networks on which applicable hosts are attached means that when such a host has a datagram destined for an IP address on the stub network, a next-hop router is known by that host which, when the datagram is sent to that router, will ultimately result in the datagram reaching the intended stub network host.

1.2. Usability Goals

In addition to the interoperability goals we’ve described above, the additional goal for stub networks is that they be able to be connected automatically, with no user intervention. The experience of connecting a stub network to an infrastructure should be as straightforward as connecting a new host to the same infrastructure network.

1.3. State of the Art

Currently there is one known way to accomplish what we are describing here [[Michael, does ANIMA have a second way?]]. The Homenet working group produced a protocol, HomeNet Configuration Protocol (HNCP), the purpose of which is to allow a collection of routers to self-configure. HNCP is not technically constrained to home environments; in principle, it can work in any environment.

The problem with HNCP is twofold. First, it only works if it is deployed on all routers within the network infrastructure for a site. Secondly, it attempts to do too much, and invents too much that is new. Let’s look at these in order.

First, HNCP only works when deployed on all routers within the network infrastructure. To be clear, this does not mean that it is impossible to use HNCP on a network where, for instance, the edge router(s) do not support HNCP. What it does mean is that if this configuration works, the reason it works is that the network supports prefix delegation to routers inside the network. So a router doing HNCP can get a prefix using prefix delegation from, for example, an edge router, and this will work.

Unfortunately, the way that such an HNCP server should behave is not documented, and it’s not actually clear how it should behave. What if the DHCP server allocates it a /64? HNCP is designed to get a larger prefix and subdivide it-there is no provision for requesting multiple delegations. So if we wanted to use HNCP to solve this problem, we would need to do additional work.

Secondly, HNCP tries to do too much, and invents too much that is new. HNCP is a complicated protocol for propagating network configuration information in a mesh. It does not assume that any network is a stub network, and because of that, using it to support stub networks is needlessly complicated.

Despite having been an IETF proposed standard since 2016, and having been worked on for quite some time before that, it is not possible to purchase a router that implements HNCP. There exists a prototype implementation in OpenWRT, but getting it to actually work is problematic, and many problems have been left unsolved, and would be quite difficult to solve with additional standards work.

We know this because several participants in the Homenet Working Group have tried to implement make it work, and yet as yet we have made no documentable progress, and indeed the Homenet Working Group appears to be on the verge of closing.

Because of the first point-the utter lack of commercial implementations of HNCP-any stub network solution that is intended to be deployed to arbitrary networks can’t rely on the availability of HNCP. This may come in the future, but is not available now, and may never be. Therefore, whatever approach is taken MAY use HNCP if available, but MUST work without HNCP. Therefore, using HNCP represents additional implementation complexity; whether this is worth doing is something that should be considered, but because using HNCP is necessarily optional, it probably makes the most sense to assume that any functionality provided by HNCP will be external to the stub network router, and that the stub network router itself need not participate in the HNCP mesh.

2. Possible Approaches

2.1. Proxy ND

2.1.1. Reachability

Proxy Neighbor Discovery provides reachability to hosts on the stub network by simply pretending that they are on the infrastructure network. This reachability can be local or global depending on what IPv6 service (if any) is available on the infrastructure link. The use of Proxy ND for providing connectivity to stub networks is described in [I-D.ietf-6lo-backbone-router].

2.1.2. Addressability

If IPv6 service is available on the infrastructure link, this service can be used to provide addressability on the stub network, and also provides addressability on the infrastructure link.

If IPv6 service is not available on the infrastructure link, addressability for proxy ND can be provided by advertising an on-link autoconfigurable prefix in a Router Advertisement offered by the stub router.

2.1.3. Discoverability

Discoverability for stub network hosts can be provided using DNS-SD service registration protocol on the stub network, in combination with an Advertising Proxy on the stub router which would advertise registered services to the infrastructure link.

Discoverability of infrastructure link hosts by stub network hosts can be provided using a DNS-SD discovery proxy and/or regular DNS. As long as the stub network requires that each stub router provide a DNS-SD Discovery Proxy and also provide name resolution, this will work even in the multiple stub router case.

2.1.4. Requirements

- The infrastructure must either provide IPv6 service, or not block the provision of IPv6 service by the stub router.

- Hosts on the infrastructure link must support IPv6 and must support IPv6 neighbor discovery.

- Every stub host must register with at least one stub router that will do proxy ND for it.

- Routers must share proxy ND information, or else each router is a single point of failure for the set of hosts that have registered with it.

- Sharing proxy ND information requires new protocol work

2.1.5. Observations

Can definitely work in specific circumstances, but probably doesn’t lend itself to full automation.

2.2. Stub reachability using RA

2.2.1. Reachability

Reachability to the stub network is provided using the Route Information Option [RFC4191] in a router advertisement [RFC4861] issued by the stub router. Since the stub router does not provide IPv6 connectivity, it must not advertise itself as a default router. Each stub router can provide a default route to the stub network.

2.2.2. Addressability

Addressability on the stub network is provided using a ULA prefix generated by the stub router. Addressibility on the infrastructure link is either provided by the infrastructure, or else must be provided by the stub router.

2.2.3. Discoverability

Discoverability for this approach is the same as for the Proxy ND approach.

2.2.4. Requirements

- Infrastructure network must not block router advertisements.

- Hosts on the infrastructure network must support IPv6, must support the use of non-default routes as described in [RFC4191], and must support routing through non-default routers (routers with a router lifetime of 0).

- Stub routers must cooperate with other stub routers in announcing an on-link prefix to the stub network.

- Stub routers must cooperate with infrastructure routers in announcing an on-link prefix for the infrastructure network. Stub routers must not advertise an on-link prefix when an on-link prefix is already present.

2.2.5. Observations

This option has the advantage of relying primarily on ordinary IPv6 routing, as opposed to workarounds like proxy neighbor discovery or NAT64. The cooperation that is required between stub routers is minimal: they need simply minimize the advertising of redundant information. When redundant information is advertised, this is an aesthetic issue rather than an operational issue, and can be allowed to heal gradually.

Additionally, this option does not require any new behavior on the part of existing hosts or routers. It does assume that infrastructure hosts actually implement [RFC4191], but it is not unreasonable to expect that this either is already the case, or can easily be accomplished. It also assumes that the infrastructure does not enforce RA Guard [RFC6105]. This is compatible with the recommendations in RFC6105, which indicates that RA guard needs to be configured before it is enabled.

The approach described in this section only makes it possible for stub network hosts to interoperate with hosts on the link to which the stub router is directly attached. The "Global Reachability" approach talks about how to establish interoperability between stub network hosts and hosts on links to which the stub network is not directly attached.

2.3. Global reachability

Global reachability for stub networks requires either the use of NAT64, or else the presence of global IPv6 service on the link. As such it is more of an add-on approach than a different approach. This section talks about a specific example of global reachability: how to make global reachability work for the "Stub Reachability using RA" approach mentioned earlier.

The "global reachability" approach has applicability both in the literal sense, and also in the sense of "reachability beyond the link to which the stub router is directly attached." The behavior of the stub router is the same in both cases: it is up to the network infrastructure what prefix is delegated to the stub router, and what reachability is provided.

2.3.1. Reachability

Reachability in this case requires integration into the routing infrastructure. This is most easily accomplished by having the DHCPv6 prefix delegation server add an entry in the routing table pointing to the stub router to which the prefix has been delegated. Stub routers can also advertise reachability to the stub network using router advertisements, but these will only work on the local link.

2.3.2. Addressability

Addressability in this case for hosts on the infrastructure link is assumed to be provided by the infrastructure, since we are relying on the infrastructure to provide DHCPv6 prefix delegation. Addressibility on the stub network is provided using the prefix acquired with prefix delegation.

2.3.3. Discoverability

Discoverability for devices on the link to which the stub network is attached can be done as described earlier under the "Proxy ND" approach.

2.3.4. Requirements

- Infrastructure network must support prefix allocation using DHCPv6 prefix delegation.

- Infrastructure network must install routes to prefixes provided using DHCPv6 prefix delegation.

- In the case of multiple stub routers, stub routers must cooperate both in acquiring and renewing prefixes acquired using prefix delegation. Stub routers must communicate complete routing information to the DHCPv6 prefix delegation server so that it can install routes.

2.3.5. Observations

This approach should be a proper superset of the "Stub Reachability using RA" approach. The primary technical challenge here is specifying how multiple stub routers cooperate in doing prefix delegation.

2.4. Support for IPv4

This document generally assumes that stub networks only support IPv6. Bidirectional reachability for IPv4 can be provided using a combination of NAT44 and Port Control Protocol [RFC6887]. The use of NAT44 and PCP in this way has already been solved and need not be discussed here.

2.4.1. Reachability

Reachability is complicated for NAT64. Typical NAT64 deployments provide reachability from the stub network to the rest of the Internet, but do not provide reachability from the rest of the internet to the stub network. As with NAT44 and PCP, this type of reachability is a solved problem and need not be discussed here. To provide complete reachability to the IPv4 internet, a stub router must not only provide reachability to the cloud, but also reachability from the cloud. That additional work is discussed here.

To provide reachability from the cloud to devices on the network, devices on the network will need to obtain static mappings from the external IPv4 address and a port to the internal IPv6 address and a port. There are three ways to do this:

* The stub host can use Port Control Protocol to register a port, and then advertise that using SRP.

* The stub host can simply register using SRP, and then SRP can establish a port mapping.

The first option has the advantage that the stub host is in complete control over what is advertised. However, it places an additional burden on the stub host which may not be desirable: the host has to implement PCP and link the PCP port allocation to the SRP registration.

For a constrained network device, it is most likely preferable to combine the two transactions: the SRP server can receive the registration from the stub host and acquire a PCP mapping for it, and then register an AAAA and A record for the host along with an SRV record for the IPv4 and IPv6 mappings. The hostname mapping would need to be different for the A record and the AAAA record in order to avoid spurious connections to the IPv4 port on the IPv6 address and vice versa.

2.4.2. Addressability

Addressability on the stub network can be provided using a ULA prefix specific to the stub network or, if NAT64 is being used in addition to one of the other solutions discussed here, the prefix allocated on the stub network for that purpose can also be used for NAT64.

IPv4 addressability on the infrastructure network is provided by the infrastructure network. It is also possible that the infrastructure network is an IPv6 network. In that case, the NAT64 edge router may be provided by the infrastructure as well.

2.4.3. Discoverability

The discoverability described for the "ND Proxy" approach should work here as well, except for the caveat mentioned above under "reachability".

2.4.4. Requirements

* TBD

2.4.5. Observations

Support for NAT64 may be required for some deployments. NAT64 support requires either close cooperation between stub routers, or else requires that the NAT64 translation be done externally. The latter choice is likely quite a bit easier; solutions that provide load balancing and high availability are already available on the market, and hence do not require that the stub routers perform this function. This is expected to be the best approach to serve the needs of consumers of this capability.

3. Discoverability Options

We can divide the set of hosts needing to be discovered and the set of hosts needing to discover them into four categories:

* Stub network hosts (stub hosts)

* Hosts that are on the link to which the stub network is directly connected (direct hosts)

* Hosts that are on other links within the same infrastructure (infrastructure hosts)

* Hosts that are on other links not within the same infrastructure (cloud hosts)

To enable stub hosts to discover direct hosts, a Discovery Proxy [RFC8766] can be used. This must be resident on any stub network router that is seen by the stub host as a resolver.

To enable stub hosts to discover infrastructure hosts using DNS-SD [RFC6763], the infrastructure must provide support for RFC6763 service discover using DNS.

To enable stub hosts to discover infrastructure hosts and cloud hosts using DNS, DNS resolution must be provided by the stub router, and the infrastructure must additionally provide the stub router with the ability to resolve names.

To enable direct hosts to discover stub hosts, stub routers must implement a DNS-SD Advertising Proxy. Stub hosts must register with the advertising proxy using SRP.

To enable infrastructure hosts to discover stub hosts, stub routers must provide authoritative DNS service for the stub network link so that it can be integrated into the infrastructure DNS-SD service. To do this automatically will require additional protocol work.

To enable cloud hosts to discover stub hosts, stub hosts would need to register with the DNS, and the infrastructure would need to make those registrations available globally, perhaps with whitelisting. This is probably not a very widely applicable use case, and we do not consider specifying how this works to be part of the work of this document.

4. Multiple Egress, Multiple Link

In the case of a stub network that has multiple stub routers, it is possible that, either when the stub network is initially set up, or subsequently, one or more stub routers might be connected to a different infrastructure link than one or more other stub routers. There are two viable approaches to this problem:

- declare it out of scope and have the stub routers prevent such configurations

- make sure that stub routers attached to each infrastructure link provide complete service on that link

Explain further.

5. Management Considerations

TBD

6. Privacy Considerations

In the locally reachable case, privacy is protected in the sense that names published locally are only visible to devices connected locally. This may be insufficient privacy in some cases.

In the globally reachable case, discoverability has privacy implications. Unfiltered automatic discoverability is probably not a good idea in the globally reachable case. If automatic discoverability is provided, some filtering mechanism would need to be specified.

7. Security Considerations

TBD

8. IANA considerations

No new actions are required by IANA for this document.

9. Informative References

[RFC4191] Draves, R. and D. Thaler, "Default Router Preferences and More-Specific Routes", RFC 4191, DOI 10.17487/RFC4191, November 2005, <https://www.rfc-editor.org/info/rfc4191>.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <https://www.rfc-editor.org/info/rfc4861>.

[RFC6105] Levy-Abegnoli, E., Van de Velde, G., Popoviciu, C., and J. Mohacsi, "IPv6 Router Advertisement Guard", RFC 6105, DOI 10.17487/RFC6105, February 2011, <https://www.rfc-editor.org/info/rfc6105>.

[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>.

[RFC6887] Wing, D., Ed., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", RFC 6887, DOI 10.17487/RFC6887, April 2013, <https://www.rfc-editor.org/info/rfc6887>.

[RFC8766] Cheshire, S., "Discovery Proxy for Multicast DNS-Based Service Discovery", RFC 8766, DOI 10.17487/RFC8766, June 2020, <https://www.rfc-editor.org/info/rfc8766>.

[I-D.ietf-6lo-backbone-router]
Thubert, P., Perkins, C. E., and E. Levy-Abegnoli, "IPv6 Backbone Router", Work in Progress, Internet-Draft, draft-ietf-6lo-backbone-router-20, 23 March 2020, <https://datatracker.ietf.org/doc/html/draft-ietf-6lo-backbone-router-20>.

Author's Address

Ted Lemon
Apple, Inc.
One Apple Park Way
Cupertino, California 95014
United States of America
Email: mellon@fugue.com