Automatically Connecting Stub Networks to Infrastructure

Ted Lemon <mellon@fugue.com>
Goals

This is the basic Stub Networks document:

• Does not depend on infrastructure support
• No host changes on infrastructure
• Host changes optional on stub network, depending on target use case

As discussed in the charter, goals are:

• Hosts on infrastructure and stub can discover each other
• Hosts on infrastructure and stub can initiate communication with each other
• Hosts on stub network can initiate communication with hosts on internet
Original document covered:

- Maintain usable IPv6 prefix on adjacent infrastructure
- SRP registration server for stub network
- Advertising Proxy advertises SRP on infrastructure
- NAT64 for cloud connectivity
- DHCPv6 for cloud/non-adjacent infrastructure connectivity
- Not much detail about discoverability across non-adjacent infrastructure links
Changes from previous

Remove text that relies on infrastructure support
  • No support for DHCPv6
  • No support for reachability from non-adjacent links

Some text from the problem statement document has been incorporated into the introduction

State machine described for managing infrastructure on-link autoconfiguration prefix
What is actually covered now

Maintenance of usable prefix on adjacent infrastructure link
Maintenance of Off-Stub-Network-Routeable (OSNR) prefix
Advertising routes to stub network
  • missing: doesn’t actually say to forward
Advertising routes to adjacent infrastructure prefix on stub
  • This is somewhat architecture-dependent
Service Advertising and Discovery
  • SRP on stub network for advertising stub network hosts
  • DNSSD-over-DNS through stub network resolver for discovery on stub network
    • Discovery Proxy for discovering adjacent infrastructure hosts
    • DNS authoritative zone for discovering stub network hosts
  • mDNS for service discovery on adjacent infrastructure
    • advertising proxy provides answers for services on stub network
    • infrastructure service discovery using regular mDNS
Maintenance of NAT64 prefix
Discussion so far

The question was raised whether we should do NAT64 in this document

Reasons not to:

• Preference not to support IPv4
• It’s not pretty

Reasons to:

• Stub network hosts need internet connectivity to:
  • Download firmware updates
  • Talk to cloud services
• We can’t count on availability of IPv6 connectivity
• Even with IPv6 connectivity, we can’t count on being able to get a delegated prefix
• This document doesn’t currently talk about prefix delegation
• So without NAT64, we can’t actually communicate with the cloud
Do we want Prefix Delegation in basic document?

I took this out—it’s a bit complicated, and does require infrastructure support.

On the other hand, in principle infrastructure could support it, and some existing home routers do.

Required for IPv6 end-to-end.

Wouldn’t be that hard to add.

What does the WG think?
Remaining work to finish

We don’t have a state machine for NAT64
Discoverability relies on draft-ietf-dnssd-advertising-proxy
  • Do we need to say more than that document will say?
We talk about partitioning, but don’t really say much
  • Do we need to say more?
The text about reachability doesn’t contain any explicit instructions.
  • Do we need to be more explicit, or is the text as written sufficient?
Working Group Adoption

Does the working group think this is the document we should adopt?

- Let’s ask

Can some folks read the document and see if it says enough that they think they could do an implementation?

- We have two implementations for Thread, but they predate the document, so that might not mean much

- Would really help to have someone read the document who hasn’t done an implementation but could; even if they don’t do an actual implementation, thinking about what they’d do might surface omissions or errors

- Would be even better if someone (Pascal?) was interested in doing an implementation for their specific application