Making Unicast DNSSD Real

Ted Lemon <mellon@fugue.com>
State of the Art

- **Manual unicast DNSSD**
  - `dig _ipp._tcp.meeting.ietf.org ptr`
  - `_ipp._tcp... 1 IN PTR reg-printer._ipp._tcp.meeting.ietf.org`
  - `_ipp._tcp... 1 IN PTR ietf114-printer._ipp._tcp.meeting.ietf.org`

- **Discovery Proxy**
  - `dig @office.local _ipp._tcp.default.service.arpa. ptr`
  - `_ipp._tcp... 10 IN PTR Brother HL-L2370DW series._ipp._tcp...

- **SRP -> DNS auth server**
- **SRP -> Advertising proxy (not really unicast)**
What would it take?

- Unicast DNS-SD service discoverable on network link(s)
- SRP service discoverable on network link(s)
  - As above. This is an authoritative DNS server
- DNSSD Discovery Proxy for discovering non-SRP-capable services through unicast DNS
- DNS full-service resolver that routes DNS-SD queries to the right place if there is no delegation from the root
- Clean fallback to mDNS if unicast service fails?
Deployment models

- Network environment
  - Single link (typical home network)
  - Single link with stub networks (also home network)
  - Multi-link SOHO
  - Multi-link enterprise

- Service environment
  - Service is part of infrastructure
  - Service is added to infrastructure, discovered via mDNS
Naming Models

- "local"
  - Should work for home network deployments, even with stub routers
  - No problem with fallback to mDNS, because names are the same
- per-link naming
  - May be required/preferred for some environments
    - which?
  - Fallback to mDNS needs serious thought. Maybe "local" always means “this link” even in this case, but other links are also browsable?
Obstacles

- A lot of applications specifically query .local
  - Does this mean “all locally connected interfaces, using mDNS?”
  - Does it mean “all services discoverable in the legacy browsing domain list?”
- For per-link naming, if .local also works, we have a stable name for every service, but also a varying name for every service.
  - Do we benefit by forcing use of varying name always?
  - Would that work?
  - Is it a problem if there are two names under which the same service can be discovered, but one is stable and the other isn’t?