What Needs to be Standardized for Hybrid Proxy?

Stuart Cheshire, DNSSD WG,
89th IETF, London, England, March 2014

How Services Advertise

- Services Advertise via existing mDNS query/response mechanism (RFC 6762)
  - Compatible with existing deployed devices

- Link's mDNS data is *conceptually* imported into a given DNS domain (e.g., services.example.com.)
  - Data served by a Hybrid Proxy, which is authoritative for that domain

How Clients Discover

- Domain Enumeration queries identify “useful” domains
  - User also has option of adding domains manually
  - Other autoconfiguration mechanisms possible

- Unicast DNS queries
  - Client has recursive DNS server configured
    - e.g. via DHCP option or IPv6 RA (RFC 6106)
  - DNS delegation (NS record) identifies authoritative server (i.e. hybrid proxy) for a given domain

Change Notification

- LLQ provides timely notification of changes
  - Keep UI up to date without “refresh” button
- Current UDP-based protocol
  - draft-sekar-dns-llq-01
  - Current Apple implementation assumes client needs either a public IPv4 address, or a NAT port mapping (PCP/NAT-PMP/IGD)
- A TCP-based LLQv2 protocol may be desirable

Security

- What Services are allowed to advertise?
  - Hybrid Proxy may impose filter or rules
- What are Clients allowed to discover?
  - Currently, any client can access service data
  - End-to-end security controls access to service
  - New LLQv2 protocol could offer per-client access control

Documented & Implemented

- Traditional unicast DNS (obviously)
- DNS-SD (RFC 6763) and mDNS (RFC 6762)
  - Services advertise using DNS-SD/mDNS
  - Clients query using Domain Enumeration & LLQ
- Hybrid Proxy (Markus Stenberg, and one other)

Work Items

- Standardize details of Hybrid Proxy
  - Timeout behavior
  - Name and rdata rewriting/suppression rules
- Improved TCP-based LLQ protocol
- Protocol for “merging” links
  - Different domain name for every link undesirable
  - When links are combined,
    need to handle duplicate names