Device Discovery with mDNS and DNS-SD

Ashley Butterworth
Matthew Xavier Mora
Apple Inc.

Multicast DNS (mDNS)

- DNS like operations on local link networks
- Sent via multicast UDP (224.0.0.251 or FF02::FB port 5353)
- Defines special top level domain (TLD) ".local." which is always link local
- Some responses are unicast
- Multicast requests and responses are cached and pruned to reduce network traffic

DNS Service Discovery (DNS-SD)

- Used DNS PTR, SRV and DNS TXT records
- Works on both multicast and unicast DNS
- Does not specify any new messages, op codes or types. Defines how existing should be used.
- SRV tells discoverer host and port
- TXT is optional for providing additional info

Pointer Records (DNS-PTR)

- **Pointer Records**
  - PTR records enable service discovery by mapping the type of the service to a list of names of specific instances of that type of service. This record adds yet another layer of indirection so services can be found just by looking up PTR records labeled with the service type.
  - The record contains just one piece of information, the name of the service (which is the same as the name of the SRV record). PTR records are accordingly named just like SRV records but without the instance name:
  - `<Service Type>.<Domain>`

    Here is an example of a PTR record for a print spooler named `PrintsAlot`:

    `_printer._tcp.local. 28800 PTR PrintsAlot._printer._tcp.local.`

Service Records (DNS SRV)

- ## (RFC 2782) Service Types

- Short name of protocol, **fourteen characters maximum**, conforming to normal DNS host name rules: Only lower-case letters, digits, and hyphens; must begin and end with lower-case letter or digit

- Longer, descriptive name of protocol.

- Name and email address of responsible person.

- URL giving description of protocol. (Or statement that the protocol is proprietary.)

- Primary Transport Protocol. ("_udp" or "_tcp", only needed for historical reasons, to determine correct SRV service type)

- List of defined TXT record keys (see draft-cheshire-dnsext-dns-sd.txt, Section 6) for this service or URL reference to document describing defined keys.

- Apple Examples:

  - **apple-ausend** Apple Audio Units
  - **apple-midi** Apple MIDI
  - **appletv** Apple TV
  - **appletv-itunes** Apple TV discovery of iTunes
  - **appletv-pair** Apple TV Pairing

TXT Records

- Text based string recommended to be under 1300 bytes to fit in single ethernet frame
- Each key value pair can be no longer than 255 bytes
  - can store booleans using presence or absence of key
- Key Value pairs stored as string with no null terminator
  - length byte followed by key ‘=’ and value
  - Value can be binary data

DNS-SD (Rendezvous) TXT record format

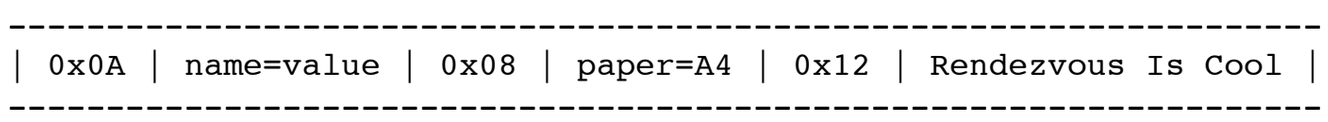
- **General format rules for DNS TXT records**
  - A DNS TXT record can be up to 65535 (0xFFFF) bytes long. The total length is indicated by the length given in the resource record header in the DNS message. There is no way to tell directly from the data alone how long it is (e.g. there is no length count at the start, or terminating NULL byte at the end).
  - The format of the data within a DNS TXT record is zero or more strings, packed together in memory without any intervening gaps or padding bytes for word alignment.
  - The format of each constituent string within the DNS TXT record is a single length byte, followed by 0-255 bytes of text data.
  - These format rules are defined in Section 3.3.14 of RFC 1035, and are not specific to DNS-SD. DNS-SD simply specifies a usage convention for what data should be stored in those constituent strings.

DNS-SD (Rendezvous) TXT record format

- **DNS TXT record format rules for use in DNS-SD (Rendezvous)**
  - DNS-SD uses DNS TXT records to store arbitrary name/value pairs conveying additional information about the named service. Each name/value pair is encoded as it's own constituent string within the DNS TXT record, in the form "name=value". Everything up to the first '=' character is the name. Everything after the first '=' character to the end of the string (including subsequent '=' characters, if any) is the value. Specific rules governing names and values are given below. Each author defining a DNS-SD (Rendezvous) profile for discovering instances of a particular type of service should define the base set of name/value attributes that are valid for that type of service. Using this standardized name/value syntax within the TXT record makes it easier for these base definitions to be expanded later by defining additional named attributes. If an implementation sees unknown attribute names in a service TXT record, it SHOULD silently ignore them.
  - The TCP (or UDP) port number of the service, and target host name, are given in the SRV record. This information — target host name and port number — MUST NOT be duplicated using name/value attributes in the TXT record.
  - The intention of DNS-SD TXT records is convey a small amount of useful additional information about a service. Ideally it SHOULD NOT be necessary for a client to retrieve this additional information before it an usefully establish a connection to the service. For a well-designed TCP-based application protocol, it should be possible, knowing only the host name and port number, to open a connection to that listening process, and then perform version- or feature-negotiation to determine the capabilities of the service instance.
- There are legacy protocols which provide no feature negotiation capability, and in these cases it may be useful to convey necessary information in the TXT record.

DNS-SD (Rendezvous) TXT record format

- **DNS-SD TXT record size**
  - The total size of a typical DNS-SD TXT record is intended to be small — 100 bytes or less.
  - In cases where more data is justified (e.g. LPR printing), keeping the total size under 400 bytes should allow it to fit in a single standard 512-byte DNS message. (This standard DNS message size is defined in RFC 1035.)
  - In extreme cases where even this is not enough, keeping size of the TXT record under 1300 bytes should allow it to fit in a single 1500-byte Ethernet packet.
  - Using TXT records larger than 1300 bytes is NOT RECOMMENDED at this time.



Device Discovery - Device

- Device publishes SRV and TXT records for service
  - mDNS/DNS-SD send gratuitous announcement after probing
- mDNS/DNS-SD responds to queries per protocol
- Device removes SRV and TXT records for service when done
  - mDNS/DNS-SD send gratuitous announcement as goodbye message (TTL of 0)

Device Discovery - Controller

- Establishes mDNS/DNS-SD query for services
- Device discovered when mDNS/DNS-SD
  - receives gratuitous announce for established query
  - receives response to query request
- Device removed when mDNS/DNS-SD
  - receives goodbye message
  - time to live expires without receiving update (timeout)

Device Discovery - Controller (cont)

- DNS Long-Lived Queries
  - Describes a protocol for setting up long-lived DNS queries with change notification, as a more efficient alternative to rapidly polling the server. (no polling)
- Dynamic DNS Update Leases
  - Describes a protocol for performing DNS Dynamic Updates with an attached lease time, that are automatically deleted unless renewed before the lease expires, much like a DHCP address lease

Device Discovery (cont)

- DNS Long-Lived Queries
- Dynamic DNS Update Leases

Bonjour

- Apple's implementation of mDNS and DNS-SD
- Available as open source under Apache 2.0 license
- Runs on Mac OS X, Window, Linux and VxWorks

Resources

- mDNS http://www.multicastdns.org/
- DNS-SD http://www.dns-sd.org/
- Bonjour http://developer.apple.com/networking/bonjour/index.html
- Bonjour dev mailing list http://www.lists.apple.com/mailman/listinfo/bonjour-dev

Other files

DNS Long-Lived Queries (draft-sekar-dns-llq.txt) describes a protocol for setting up long-lived DNS queries with change notification, as a more efficient alternative to rapidly polling the server.

- Dynamic DNS Update Leases (draft-sekar-dns-ul.txt) describes a protocol for performing DNS Dynamic Updates with an attached lease time, that are automatically deleted unless renewed before the lease expires, much like a DHCP address lease.

- NAT Port Mapping Protocol (NAT-PMP) (draft-cheshire-nat-pmp.txt) describes a protocol for asking a home NAT gateway for its "public" address, so that a host behind a NAT gateway can create a DNS Dynamic Update using that public address, rather than its less useful private address, and similarly for asking a home NAT gateway to assign a public port number and an inbound port maping, so that a host behind a NAT gateway can create DNS SRV records using that public port number rather than its less useful private internal port number.