

# Use of DNS-SD/mDNS by AllJoyn

Dave Thaler <dthaler@Microsoft.com>

Based on code analysis and information in  
<https://allseenalliance.org/framework/documentation/learn/core/system-description/advertisement-discovery-draft-aggarwal-dnssd-optimize-query-00> (expired)

# Why me? 😊

- I am not an author of any of the behavior described here
- Microsoft analyzed the behavior, so presenting what we've seen
- Original author of code and draft no longer working in this area
- We would like to improve it, and the AllSeen Alliance is open to that (but backwards interop is important too)
- We also believe that some mDNS extensions would be beneficial

# AllJoyn service discovery requirements

- Efficiently enumerate responders implementing logical functions (e.g., light, thermostat, motion sensor, etc.) called “interfaces” in AllJoyn
- Make it easy for vendors to mint unique function types at will (based on their DNS suffix)
- Support proxy responders (to non-IP links, sleepy nodes, etc.)
- Local subnet (so mDNS only), but responders may be proxying to other non-IP links
  - Only used for service discovery with shared .local names, so did not use DNSSEC
- High reliability and low latency on WiFi (but not limited to WiFi)
- Scale up to a large number of responders with a large number of function types each
- Support multiple apps per IP address
  
- AllJoyn allows multiple service discovery protocols
  - But have to worry about backwards compatibility

# Why mDNS extensions?

- “there was a brand of WiFi router very popular in Israel and another in China that was causing problems”
- Specifically, reported that AP’s would forward mDNS but *not* other multicast traffic
- So forced to use mDNS port

# Multicast over WiFi is much less reliable than unicast

- AJ sends bursts of 3 queries (100ms apart) at t=0, 1, 3, 9, 27 sec
- Responder responds to at most one query per burst
- Responses are always unicast, but only one app can use unicast per port
- AJ adds mDNS extension whereby querier can specify alternate address/port for unicast response

# DNS-SD based query

- Documented at <https://allseenalliance.org/framework/documentation/learn/core/system-description/advertisement-discovery#dns-sd-message-format>
- Name: `alljoyn._udp.local.` or `alljoyn._tcp.local.`
- Additional section: TXT records with name/value pairs
  - `search.guid.local.:` `txtvers`, `i_#` (e.g., “`i_1 = org.alljoyn.About`”) for each interface to `‘.AND’`
  - `sender-info.guid.local.:` `txtvers`, protocol version, **alternate IP addr/port to unicast response to, burst id**

# DNS-SD based response

- Additional record section
  - sender-info.*guid*.local. TXT like in query but with responder's info
    - Querier can optionally use to get alternate port for subsequent unicast queries
  - advertise.*guid*.local. TXT n\_1=org.alljoyn>About.sl.y.x
    - Deprecated, but still present for backwards compatibility
  - *guid*.local. A/AAAA address
    - Should also be deprecated

# draft-aggarwal-dnssd-optimize-query-00 (expired)

- Querier optionally puts TXT records in additional section of mDNS query
  - Filters based on TXT key/value because DNS-SD already uses TXT key/value pairs
  - Simple queries only since goal is just optimization, some extras are ok
    - reduce multicast traffic
    - reduce transmission requirements on responders
- Responder optionally uses them to filter on and decide whether to respond

# Feedback from IETF 90 minutes & jabber log

- Tim Chown said the technology described in the document may not be explicitly in the WG charter, but is important to consider, for the design tradeoffs it discusses.
- Tim Wicinski pointed out that the technology might be problematic with DNSSEC.
- Stuart Cheshire said the technology seemed similar (problematically) to SLP in that it defines a "query language".
- [\[20:51:14\]](#) <Andrew Sullivan> Clearly, if we want a query language, then we want some matching. So, what is needed is not TXT records, but NAPTR records with the full-blown regex profile in the additional section.
- Dave Robin pointed out that subtype filtering is important as hierarchical organization cannot scale.
- Kerry Lynn suggested there might be something to be learned in this area from SEP 2.0.
- There was no decision about the disposition of this document, but the chairs would like the contents discussed further, preferably on the mail list.

# Next steps

- Determine whether WG wants to work on any extensions (whether based on what AllJoyn did or something else)
  - WiFi optimizations?
  - Alternate unicast response port?
  - Simple filters in queries for shared .local names
- Resurrect expired draft (Dave Thaler willing, co-authors accepted)